if

CALCULATOARE PERSONALE

Revistă lunară editată de Micro ATCI Tîrgu Mureş

1 / 91
(4)
IANUARIE
39 Lei

if if if

Imprimanta - sau fără hîrtie nu se poate !

Utilizarea raţională a memoriei

Baze de date

Unix şi viruşii

Evenimentul PS/1

Anchetă - vreau să cumpăr un AT !

HARDWARE-UL DUMNEAVOASTRĂ ESTE TOT ATÎT DE BUN CÎT ESTE ŞI PARTENERUL DE SERVICE PE CARE L-AŢI ALES !!

SIRECA - Secţia de întreţinere şi reparaţii echipamente de calcul şi automatizări din cadrul întreprinderii INFO-HARD execută cu personalul său calificat întreţinerea şi repararea echipamentelor de calcul electronice şi automatizări din gama:

- sisteme de calcul FELIX-256-1024 , FELIX-5000;
- minicalculatoare CORAL , INDEPENDENT;
- videoterminale VDT , alfagraf;
- stabilizatoare de tensiune de 10 KVA si 20 KVA;

De asemenea , prin specialiştii din laboratorul central,execută:

- lucrări de repunere în funcţiune sisteme de calcul;
- transformări de echipamente;
- reparaţii de plachete;

Telefoanele de la sediul central şi de la filialele din ţară sînt:

-sediul central:

BUCUREŞTI , str.Cluceru Sandu Nr.2 , sector 2 , cod 73156
telefon: 90/276330-int.3
sau..... 90/273185

-puncte de lucru:

ARAD.....966/14173
BACĂU 931/42895
BAIA-MARE 994/34983
BRAŞOV 921/10430-int.226
BRĂILA 946/35863
CLUJ 951/30677
CRAIOVA 941/32934
FOCŞANI 939/22087
IAŞI 981/37947
ORADEA 991/42761
PLOIEŞTI 971/44778
PITEŞTI 976/35492
SIBIU 924/16502
TIMIŞOARA 961/16599
GIURGIU 912/15021
RÎMNICU VÎLCEA 947/11571
TÎRGOVIŞTE 926/11951-int.25

Pentru informaţii suplimentare specialiştii noştri vă stau la dispoziţie.

if

revistă de informatică
editată de firma **Micro ATCI**

Dìrector: ing. Dumitru Dunca

La realizarea acestui număr au colaborat:

ing. Attila Darvas,
ing. Zoltán Egyed,
ing. Iosif Fettich,
ing. Alin Flaidăr,
ing. Ingrid Maier,
ing. Romulus Maier,
ing. Emil Palade
Desene: ing. Ştefan Doboş

Tiparul: tipografia Tîrgu Mureş

Revista apare lunar.

Preţ: 39 lei per exemplar, la care se adaugă eventualele taxe de expediţie.

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/17024, fax 954/35208.

Manuscrise originale sau listing-uri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu directoru'. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate la următoarele tarife:
1 cm2 (alb-negru) 30 lei,
1 cm2 (alb-negru şi o culoare suplimentară) 40 lei,
1 cm2 color 60 lei.
Doritorii sînt rugaţi să ia legătura cu redacţia.

Angajament

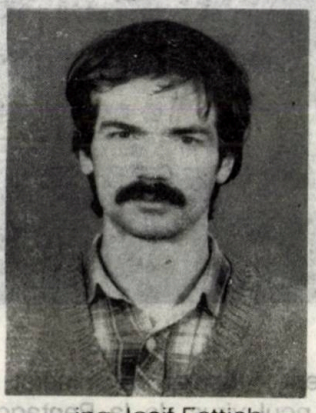
Subsemnaţii, echipa redacţională a revistei **if**, acum la început de an, ne angajăm solemn în faţa cititorilor noştri să facem totul ca în ciuda greutăţilor, piedicilor, vicisitudinilor sorţii, etc., etc., revista **if** să meargă mai departe... Piei satană !

În mod serios nu vă promitem nimic, dar vă împărtăşim cîteva dintre dorinţele şi speranţele noastre. Dorim ca revista noastră, care de-acum sperăm că este şi a d-voastră, să apară lunar. Sperăm că aşa va fi, dar nu sîntem siguri. În privinţa acestui prim număr din '91 Moş Crăciun şi Moş Gerilă au stabilit un consens şi i-au dat un brînci pe gheţuşul sărbătorilor de iarnă sporind şi mai mult întîrzierea cu care va apare pe piaţă. În momentul de faţă cea mai dificilă problemă cu care ne confruntăm s-ar putea formula în modul următor: "Dacă se predă la tipografie un material la o dată prestabilită x şi se promite tipărirea materialului pînă la data y, care este data z la care va putea fi ridicat materialul ?". Dacă ne-aţi putea comunica adresa vreunei babe care ştie să ghicească în bobi, sau în orice altceva, care ne-ar putea ajuta în această problemă, v-am fi recunoscători.

Dorim, de asemenea, ca structura şi conţinutul revistei să se îmbunătăţească şi îmbogăţească continuu. Sîntem siguri că, dacă veţi continua să ne jutaţi cu aprecierile şi criticile d-voastră, aşa se va întîmpla. Iar dacă vom mai şi greşi sperăm că ne veţi ierta. Este foarte greu să mulţumeşti pe toată lumea. Ni s-a întîmplat, şi nu o singură dată, ca un articol apreciat de mai mulţi cititori să nu placă altor cititori. Nivelul articolelor reprezintă o altă piatră de încercare. Unii cititori ne reproşează că nu "urcăm destul de sus", în timp ce alţii se plîng de faptul că nu pornim "mai de jos". Pluta conceptului redacţional pluteşte astfel între Scylla şi Charybda, dar ţinînd bine cîrma sperăm că ... "E la nave va !".

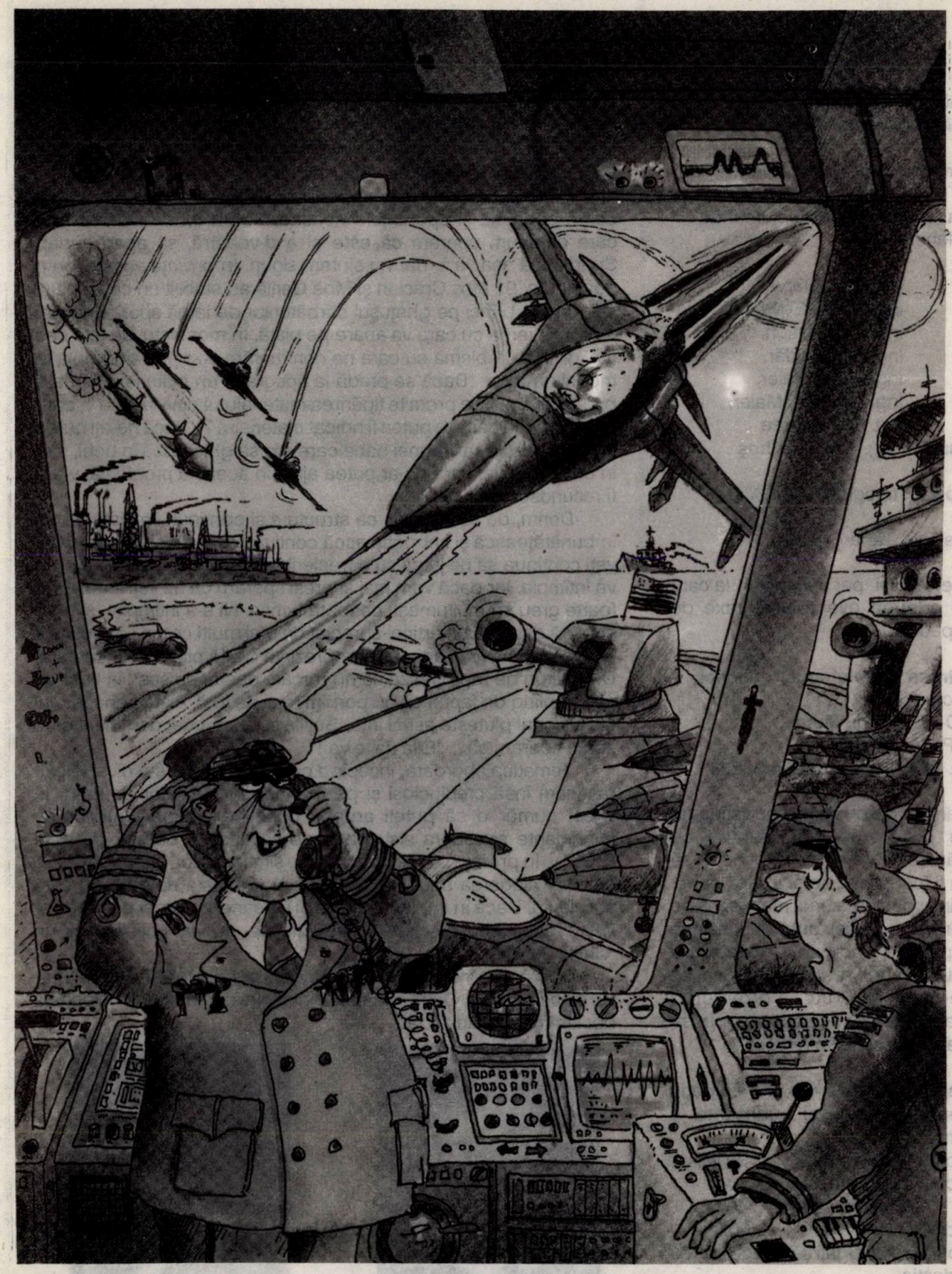
Tematica abordată, începînd cu acest număr, s-a diversificat, rămînem însă credincioşi şi pe mai departe temelor de fond. În acest număr o să puteţi arunca o privire asupra familiilor de imprimante şi asupra sistemelor de gestiune a bazelor de date (SGBD). În privinţa bazelor de date ne simţim datori să vă avertizăm că doar alegerea unui SGBD, oricît ar fi acesta de bun, nu este suficientă. Dacă în implementarea şi întreţinerea bazei de date vor lipsi organizarea, disciplina, rigurozitatea, nu veţi obţine decît un "cimitir de date".

Sperăm ca informaţiile din articolele publicate să nu vă fie doar pe plac ci şi utile. Dacă aşa se va întîmpla înseamnă că osteneala noastră nu este în zadar.



ing. Iosif Fettich

ing. Romulus Maier



Everything O.K., Mr. President - tocmai am infectat reţeaua de calculatoare adversă cu noul virus de la Pentagon ...

Cuprins

Editorial . . . . . pag. 1

Ştiri

- Trei noi PC-uri marca Apple . . . . . pag. 4
- Placă TV pentru PC-uri . . . . . pag. 4
- Cartelă PC-Fax pentru Est . . . . . pag. 4
- Afacerea Borland versus Lotus : o poveste urîtă, tristă, regretabilă ! . . pag. 5
- Specificaţii pentru administrarea LAN-urilor . . . . . pag. 5
- Standard multimedia . . . . . pag. 5
- Urmaşul lui Intel pentru SX . . . . . pag. 6
- Discuri flash-memory . . . . . pag. 6

Magazin

"Destinse, fără frică şi vesele" . . . . . pag. 7

Multimedia

A noua, vă rog ! . . . . . pag. 8

Simfonia a noua a lui Ludwig van Beethoven, în cuvinte şi imagini, cu sunet digital Hi-Fi : noul eveniment multi-media pentru Mac.

Vă prezentăm

Evenimentul PS/1 ? . . . . . pag. 10

Un microcalculator puternic la îndemîna tuturor. O victorie à la Pyrrhus pentru Apple, care-şi vede strategia convivialităţii preluată de IBM. Şi totuşi, ce aduce nou PS/1?

Virusologie

Unix şi viruşii . . . . . pag. 12

- "Orange Book"- ce înseamnă "sigur"? pag. 15

Hardware

Imprimanta - sau fără hîrtie nu se poate ! . . . . . pag. 16

- 7 sfaturi pentru o instalare corectă a imprimantei . . . . . pag. 18
- 7 sfaturi pentru achiziţionarea de imprimante. . . . . pag. 18
- Cele mai cunoscute imprimante . . pag. 19

Cum se lucrează cu Extended Memory

Utilizarea raţională a memoriei calculatoarelor personale . . . . . pag. 20

Reţele

10 Base-T - Strategii de implementare . pag. 26

10 Base-T este noul standard pentru Ethernet pe cablu bifilar neecranat. Implementarea standardului ,cu multe funcţiuni noi, este foarte simplă, dacă se respectă anumite reguli.

Software

Baze de date . . . . . pag. 29

- dBase IV 1.1 - revenire reuşită . . . pag. 30
- Paradox 3.5 - o punte între două lumi . . . . . pag. 31
- R:Base 3.1 - examen de maturitate . pag. 33

Compilatoare de baze de date . . . . . pag. 35

- Clipper 5.0 . . . . . pag. 35
- Foxpro 1.0 . . . . . pag. 36
- Clarion 2.1 . . . . . pag. 37

Aplicaţii distribuite . . . . . pag. 38

Practica

Se poate formata şi altfel... . . . . pag. 44

Pe un AT cu o unitate floppy de tip MF, utilizatorul nu poate produce pe dischete simple (2S,DD) dischete de 420 kbyte, 720 kbyte sau 800 kbyte. O alternativă o oferă programul "mformat".

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Caricatura . . . . . pag. 2
- Preţuri, preţuri... . . . . pag. 52
- Anchetă - Vreau să cumpăr un AT ! . . pag. 53
- Poşta redacţiei . . . . . pag. 54
- Mica publicitate . . . . . pag. 54

Trei noi PC-uri marca Apple

Cu cele trei noi calculatoare Macintosh prezentate: "Classic", "LC" şi "IIsi", Apple încearcă pentru prima dată să cîştige cumpărători în domeniul costurilor reduse. Apple oferă un model pentru începători, în versiunea cea mai ieftină, la acelaşi preţ la care IBM oferă modelul PS/1, adică aprox. 2000 mărci. Macintosh Classic este oferit, ca şi predecesorul Mac Plus, într-o carcasă integrată în care se află atît calculatorul cît şi monitorul. Procesorul este un Motorola 68000 la o frecvenţă de tact de 8 MHz, iar unitatea floppy standard este "Superdrive", care poate prelucra dischete de 3,5" în format MS-DOS, OS/2 şi Pro DOS (Pro DOS este formatul dischetelor vechiului Mac Plus). Sistemul mai conţine: 1 MByte memorie de lucru, şase interfeţe, facilităţi de reţea, sistemul de operare V.6.06, programe utilitare, tastatură şi mouse. O limitare o constituie lipsa slot-urilor interne, extensiile fiind posibile numai prin intermediul interfeţelor SCSI. Modelele Mac Plus şi Mac SE nu vor mai fi produse deoarece Classic urmează să le înlocuiască pe amîndouă. Classic va fi oferit şi într-o versiune cu 40 MBytes harddisk şi cu 2 MBytes RAM şi va costa 3000 mărci.

Cu o clasă mai sus se situează Macintosh LC, care a fost primul model Apple prezentat într-o carcasă desktop de 8 cm înălţime. LC are un microprocesor 68020, 16 MHz, dar nu are nici coprocesor şi nici soclu pentru acesta. Spre deosebire de Classic, LC are un slot de extensie cu acces direct la bus-ul CPU. Slotul direct al lui LC nu este totuşi compatibil cu slotul direct 68030 al lui Mac SE/30, dar ar putea fi folosit pentru o plachetă pentru un procesor 68030, care să poată utiliza modul de tratare a memoriei al de mult anunţatului sistem 7.0. Standard, cu ajutorul lui "Voice-Mail", LC este capabil să prelucreze semnale audio, preluate prin intermediul unui microfon livrat odată cu sistemul. Spre deosebire de redarea pură a sunetului în acest caz pot fi memorate documente sau ştiri (de ex. în reţea) cu comentarii şi apoi pot fi apelate. În mod standard LC dispune de: unitate floppy Superdrive, şapte interfeţe, un slot, 2 MBytes RAM, 40 MBytes harddisk, tastatură şi software sistem. Memoria de lucru poate fi extinsă pînă la 8 MBytes. Configuraţia, cu un monitor de 12", costă aprox. 4900 mărci.

Ca model de trecere spre cunoscutele calculatoare Mac II, este gîndit noul model Macintosh IIsi, care dispune de un procesor Motorola 68030, 20 MHz, şi este oferit în trei versiuni: IIsi 2/40, IIsi 5/40 şi IIsi 5/80. Cele trei variante se deosebesc doar în ceea ce priveşte capacitatea memoriei de lucru şi a harddisk-ului. La toate trei le sînt comune: o posibilitate de extindere a memoriei pînă la 17 MBytes, unitate floppy Superdrive, microfon, software sistem, ca şi opt interfeţe şi un slot pentru o plachetă Nubus sau o plachetă de extensie 030. IIsi 2/40 (2 MBytes RAM şi 40 MBytes harddisk) costă 8.000 mărci, IIsi 5/40 costă 9.000 mărci, iar IIsi 5/80 costă 10.000 mărci.

Placă TV pentru PC-uri

O firmă canadiană nouă, AView Technology, şi-a planificat introducerea pe piaţă a unei plăci pentru calculatoarele personale compatibile IBM, care să permită recepţionarea imaginilor TV pe monitorul calculatorului. Placheta "Desktop TV-Board" este o plachetă pe 8 biţi pentru monitoare multi-frecvenţă. Placheta va putea fi activată, prin intermediul unui soft propriu, sub sistemul de operare DOS, sau Windows, va putea prelucra semnale după standardul american NTSC, dar şi semnale video, şi va permite comutarea între modurile de lucru televizor şi monitor PC. Prin intermediul software-ului sonorul, contrastul, luminozitatea şi selecţia canalului, vor putea fi comandate de la tastatură. Prima versiune a plachetei lucrează numai cu monitoare multi-frecvenţă în domeniul 15,7 pînă la 31,5 kHz şi numai cu plăci grafice EGA şi VGA. O versiune ulterioară va lucra cu monitoare VGA pure şi cu plăcile grafice corespunzătoare. Ambele plăci vor costa în jur de 395 de dolari şi vor fi disponibile începînd din februarie.

Cartelă PC-Fax pentru Est

Cartela pentru PC-uri "JT-Fax", a producătorului Interquad, a fost concepută în mod special pentru liniile telefonice proaste din Germania de Est. În plus cartela de 8 biţi pentru PC-u i IBM XT/AT, este livrată cu un adaptor extern pentru operarea fără probleme pe PC-urile Robotron de 16 biţi Eser EC1834. Cartela fax respectă normele grupei III după CCITT (pînă la 4800 caractere/secundă, cu autofallback pînă la 1200 de bauds) şi costă inclusiv software-ul, manualul şi cablul de cuplare la telefon 799 mărci, fără impozitul pe valoarea adăugată. Pentru desfacerea plăcii răspunde firma Beeking Consulting din Kranenburg. Cartela JT-Fax a primit avizul de utilizare la 24.8.90 din partea Institutului pentru poştă şi medii de comunicare la distanţă, din acea vreme.

Afacerea Borland versus Lotus : o poveste urîtă, tristă, regretabilă !

Informatica nu mai este de mult domeniul programatorilor izolaţi. Încetul cu încetul geniile în micro au cedat locul oamenilor de afaceri. Telefoanelor de felul " Hey man ! Noaptea asta am găsit o chestie grozavă..." le urmează înregistrările legale, procesele verbale, certurile avocaţilor .

Filmul afacerii: Joi 28 iunie 1990, la Federal District Court din Boston se stabileşte că societăţile Paperback Software şi Stephenson Software Ltd au încălcat drepturile de copyright ale pachetului Lotus 1-2-3 în comercializarea produsului lor VP-Planner. Vineri 29 iunie 1990, Borland depune plîngere împotriva lui Lotus la District Court, California de Nord, cu scopul de a obţine o hotărîre judecătorească declarativă specificînd că pachetele de programe de calcul tabelar Quattro si Quattro-Pro nu contravin drepturilor de copyright ale lui Lotus . Luni 2 iulie, Lotus angajează o acţiune juridică în statul Massachussetts împotriva lui Borland pentru violare de copyright . Lotus pretinde că interfeţele utilizator ale lui Quattro încalcă copyright-ul lui Lotus 1-2-3 . Lotus îşi explică poziţia prin Thomas Lemberg, vice-preşedinte şi secretar general al firmei: "Lotus 1-2-3 este produsul nostru cel mai important, în care am investit resurse considerabile pentru satisfacerea necesităţilor crescînde ale clienţilor noştri. Faţă de clienţii noştri, faţă de angajaţii şi acţionarii noştri avem responsabilitatea de a ne proteja produsul împotriva oricărei copii ilicite. Noi considerăm în special că dezvoltînd Quattro, Quattro Pro şi SCO Professional, Borland International şi Santa Cruz Operation au copiat comenzile şi structura meniurilor lui Lotus 1-2-3 practic in totalitate ." De cealaltă parte Borland afirmă că Quattro Pro este un program original şi nu contraface în nici un caz Lotus 1-2-3 . În ceea ce priveşte presa americană, ea a adoptat o poziţie mai degrabă severă în ceea ce-i priveşte pe protagoniştii afacerii : " Quattro Pro nu copiază Lotus în maniera lui Paperback, dar oferă efectiv posibilitatea utilizării de meniuri şi de comenzi Lotus 1-2-3 ca o alternativă la propriile sale comenzi . Numeroşi critici consideră Quattro un program superior lui 1-2-3 ; face tot ceea ce face şi Lotus, dar are nevoie de mai puţină memorie, are grafice mai bune şi este efectiv vîndut la un preţ de 99$ spre deosebire de 495$ pentru Lotus 1-2-3 ." Jim Seymour, PC Week, 16/07/90

" Dacă nu puteţi cîştiga competiţia prin preţ, calitate, inovaţie sau servicii puteţi întotdeauna recurge la o ultimă posibilitate: angajaţi un avocat şi prezentaţi-vă în faţa instanţei. Este ceea ce se învaţă acum în şcolile de comerţ, este lecţia de care şi-a amintit Lotus cînd s-a găsit în faţa unei competiţii dure cu admirabilul nou program de calcul tabelar al lui Borland, Quattro Pro ." T. R. Reid, Washington Post, 16/07/90

"Se atinge uneori un stadiu cînd protejarea soft-ului pretinde mai multe eforturi decît cele necesare pentru a crea ceva nou. Lotus a atins acest stadiu şi l-a şi depaşit, presupunînd lesigur că i-au rămas destule resurse pentru a face ceva nou. Judecînd însă după cele petrecute, s-ar părea că această ultimă prezumţie nu este credibilă. " Jerry Pournelle, Info World, 23/07/90

Este dificil de înţeles un proces pentru o interfaţă, într-un moment în care piaţa se deplasează spre o uniformizare a meniurilor în beneficiul utilizatorului. Din fericire cei doi protagonişti nu şi-au portat încă produsele sub Windows. În ziua cînd o vor face riscă să devină pîinea avocaţilor. Cel puţin se poate spune că "afacerea" provoacă o adevarată huiduială în presa americană. Oamenii celor două societăţi beneficiază de o publicitate dificil de obţinut în condiţii normale. Recurgerea la justiţie prezintă astfel noi virtuţi .

Specificaţii pentru administrarea LAN-urilor

IBM şi 3Com au anunţat că specificaţiile elaborate împreună pentru administrarea reţelelor locale (LAN) au devenit publice. Aducerea la cunoştiinţa publică a fost precedată de o evaluare a lor din partea echipelor tehnice ale firmelor Banyan, Microsoft, Novell şi SCO. Specificaţiile pentru administrarea LAN-urilor în medii eterogene oferă structura de bază pentru elaborarea unor produse care să poată lucra cu o multitudine de sisteme de operare. Sînt disponibile şi diferite interfeţe pentru comunicaţia între sisteme de operare şi adaptoare de reţea. Interoperabilitatea, scopul final al specificaţiilor, este un element cheie al prelucrării automate a datelor în viitor.

Standard multimedia

IBM şi Microsoft, împreună cu alţi şapte producători, doresc să impună un standard comun pentru tehnologia multimedia. Multimedia trebuie să contopească într-un tot unitar domenii diferite, cum ar fi: muzică, video, calculatoare sau televiziune. Observatorii pieţei aşteaptă de la această tehnologie un nou impuls de creştere pentru industria de calculatoare a viitorului. IBM a trasat ca urmare primele linii directoare producătorilor, pentru proiectarea de software multimedia sub OS/2 şi Windows 3.0. După cum anunţă IBM specificaţiile prevăd un format general valabil de conversie fişiere pentru imagine, text şi prelucrarea sunetului ca şi o standardizare a componentelor hard corespunzătoare. Michael Braun, vicepreşedinte la IBM este optimist: "Noi credem că multimedia este o mare provocare şi şansă pentru industria calculatoarelor".

Urmaşul lui Intel pentru SX

Intel, SUA, a creat un nou microprocesor, "386SL", destinat în mod special calculatoarelor portabile şi altor calculatoare de format mic. După datele furnizate de Intel, urmaşul lui 80386SX este mai rapid şi este dotat cu mai multe funcţii.

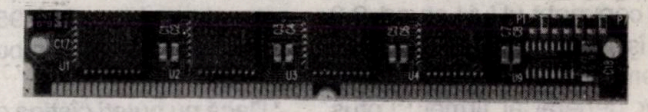
Noul microprocesor "386SL", şi chip-ul de I/O corespunzător "82360", sînt produse numai în versiunea 20 MHz şi concurează chiar şi cele mai rapide microprocesoare "386SX" produse de Intel. Desigur, Intel a integrat în acest procesor şi funcţii suplimentare. Acesta a fost realizat cu un nou sistem de comandă a întreruperilor (System Management Interrupts - SMI), un nivel de întreruperi pur hard cu un nou spaţiu de adresare. Modul de lucru SMI este invizibil pentru sistemul de operare şi pentru aplicaţiile soft şi controlează operaţiile interne, periferia, viteza CPU şi funcţiile de protecţie împotriva căderilor de curent. Intel afirmă că, în ciuda funcţiilor suplimentare, care necesită şi ele timp, un 386SL este mai rapid decît un 386SX la acelaşi tact de 20 MHz. Procesorul SL dispune de un controller suplimentar de memorie cu un spaţiu de adresare de 32 MBytes, un LIM/EMS 4.0, bus AT şi controller cache şi susţine coprocesorul 387SX. Intel livrează procesoarele 386SL şi 82360 şi separat. Preţurile, pentru o comandă de 100 de bucăţi se ridică la 176 dolari pentru 386SL şi 45 dolari pentru 82360.

Setul de procesoare 386SL şi 82360 îi pune în dificultate pe ceilalţi producători, care oferă funcţiile suplimentare ca şi componente proprii. Pentru moment doar Intel oferă un procesor care să aibă aceste funcţii integrate. După cum afirmă Intel această soluţie este mai bună decît cele oferite de concurenţă, deoarece în cazul lui SL procesorul nu mai este încărcat cu sarcini suplimentare de supraveghere a memoriei şi a întreruperilor şi de coordonare cu sistemul de operare şi cu aplicaţiile, ci aceste funcţii sînt integrate şi nu mai au nevoie de control extern.

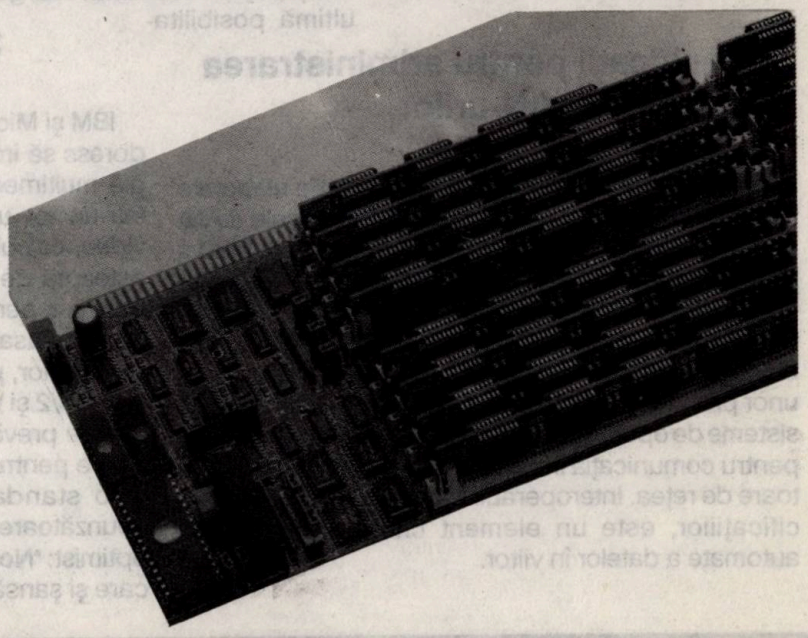
Discuri flash-memory

Dezvoltări noi în tehnologia chip-urilor de memorie au făcut posibilă apariţia discurilor flash-memory. Firma SCM (SCM Microsysteme GmbH, Bertha-v.-Suttner-Weg 1, D-8033 Martinsried b. München, Tel. 089/8561363, fax 089/8561412) prezintă primele flash-memory discuri de 16 Mbyte (utilizînd module SIMM de 1 Mbyte) sau 32 Mbyte (utilizînd module SIMM de 2 Mbyte). Un flash-memory disc este de fapt o plachetă (cartelă), PC-XT şi AT compatibilă, folosibilă cu un bus de tip ISA sau EISA, putînd opera transferuri pe 8 sau 16 biţi, dotată cu 16 socluri SIMM, constituindu-se într-un "hard-disk" de memorie nevolatilă.

Flash-diskul de memorie nevolatilă face posibilă o mărire a vitezei de 100 de ori (!) în comparaţie cu un harddisk obişnuit. Deoarece nu există uzură mecanică, şi durata de viaţă creşte cu ordine de mărime, astfel încît devine posibilă utilizarea în aplicaţii industriale, unde apar praf, vibraţii şi temperaturi ridicate. Un driver DOS (soft) face posibilă o integrare simplă în PC. E disponibilă şi o versiune autoboot (de pe care se încarcă direct sistemul la pornire). Flash-diskul poate fi îndepărtat din sistem fără ca datele stocate pe (în?) el să sufere; cu toate acestea, nu este necesară nici un fel de baterie. Flash-diskul poate fi citit/scris de sub DOS şi emulează un harddisk. EPROM-discul - altfel întrutotul similar flash-diskului - poate fi programat (scris) numai o dată. O nouă programare este posibilă numai prin schimbarea modulelor SIMM (ceea ce oferă, pe de altă parte, o excelentă protecţie împotriva viruşilor).



Modul SIMM de 2 Mbyte, scara 1:1



"Destinse, fără frică şi vesele"

În München s-a înfiinţat prima şcoală de calculatoare pentru femei- fără bărbaţi, căci "chinezeasca" lor în calculatoare deranjează climatul de studiu.

"Emancipare! Acesta a fost primul cuvînt pe care fiul meu l-a citit din acest program". Tînărul de 18 ani a răsfoit programa şcolii de calculatoare pentru femei, unde mama lui a petrecut sfîrşitul de săptămînă - trei ore vineri şi cîte opt ore sîmbătă şi duminică. Pe programă era scris: "Bazele prelucrării electronice a datelor + MS-DOS (inclusiv ergonomie practică)".

În martie 1990 a apărut ideea deschiderii unei şcoli comerciale de calculatoare pentru femei. "Nu am vrut să dezvolt un proiect model, la care an de an trebuie să alergi după mijloace de transport", povesteşte Angelika Huber faza de început. Deci s-a înscris la un curs de întreprinzătoare şi a ajuns la concluzia: "sau e greu sau e foarte greu". A înălţat în cîteva luni şcoala din pămînt. În august a preluat încăperile, în septembrie s-a sărbătorit inaugurarea.

Intreprinzătoarea în vîrstă de 30 de ani este de profesie comerciant bancar. Ea a predat generaţiei tinere de bancheri, apoi a studiat Germanistica, s-a lansat ca autoare, şi a predat în diferite instituţii cursuri de prelucrare electronică a datelor. Aici a fost deranjată de faptul că, deşi femeile sînt cele care vin în primul rînd la cursurile de prelucrare electronică a datelor, acestea nu sînt concepute pentru femei. "Cîteodată la un curs aveam cîte 15 femei la un bărbat. Chiar şi în acest raport bărbatul a reuşit să domine cursul cu cunoştiinţele lui şi să-mi vîre pe gît o discuţie despre un 386 SX". Această experienţă regretabilă a fost confirmată şi de cercetarea efectuată în cadrul cursurilor de calculatoare ţinute de femei pentru femei: aici printre altele se afirmă, că toate participantele consideră că o învăţare efectivă se poate obţine doar pornind de la o atmosferă "lejeră, fără frică şi veselă". În continuare studiul arată că motivul acestei atmosfere de lucru pentru multe cursante era faptul că "la acest curs au participat doar femei". Urmarea: "Au considerat cursul ca un spaţiu liber, unde au putut învăţa fără stress şi fără să fie mînate de la spate de un bărbat, care din principiu le ştie pe toate mai bine".

La fiecare sfîrşit de săptămînă opt femei stau în încăperea de şcolarizare. Fiecare are propriul ei loc la calculator. "Femeile grase trebui să stea la perete", se distrează o participantă de locul îngust al suprafeţelor de pe mesele de calculator. "Locul este prea mic pentru un bloc-notes", observă alta. Lipseşte şi podeaua antistatică. Cum trebuie să arate un loc de muncă atrăgător pentru om, nu pentru tehnică? Ce şanse sînt de a se împotrivi anumitor pericole? Se face un schimb de informaţii. "Deunăzi am citit un articol interesant pe această temă. Am să vi-l trimit"- femeile nu se tutuiesc.

Şase din cele opt cursante nu au mai lucrat niciodată pe un calculator înainte de acest sfîrşit de săptămînă. Două eleve au mai participat la alte cursuri şi le pare rău şi în ziua de azi: "am petrecut un week-end întreg acolo şi în final n-am ştiut nici cum se introduce o dischetă în unitate" ne spune una dintre ele. Cealaltă a plătit pentru un week-end aproape 1000 de mărci şi a fost înscrisă la un curs, care făcea trecerea pentru specialişti de la Word 4.0 la 5.0 ,"cu toate că ştiau că sînt începătoare". N-a priceput nimic, dar a fost o învăţătură pentru ea.

Doar în München aproximativ 180.000 de locuri de muncă la calculator sînt ocupate de femei. Dar Irmgard Kerschbaum, care se ocupă la Siemens filiala din München de egalitatea şanselor femeilor în meserie, este de părere că aceste cursuri pentru femei "au puţin sens". Cu toate că această problemă se discută mereu, noi "nu oferim asemenea cursuri". Motivaţia: cursurile pentru femei nu sînt rentabile. În cadrul filialei germane a celui mai mare producător de calculatoare IBM modul diferit de apropiere de tehnologia calculatarelor este considerată ca o problemă specifică vîrstei şi nu sexului.

Referentul de presă Cornelia Weiss consideră cursurile pentru femei ca o adevărată "discriminare a femeilor".Tocmai această şcolarizare separată va întări ideea preconcepută a bărbaţilor, cum că femeile ar fi mai proaste.

Elevele şcolii de calculatoare pentru femei discută această problemă foarte des. Întrebarea îngrijorătoare apare des: oare acest certificat va fi recunoscut în cadrul solicitărilor de angajare? "Dacă se leagă cineva de faptul că în certificat scrie ceva despre femei, atunci asta este problema sa şi nu a mea", obiectează o cursantă. În schimb, atunci cînd şeful de personal pune problema emancipării specialiştilor de calculatoare, femeia va trebui să renunţe la job. Angelika Huber vede soluţia acestei întrebări mai degrabă pragmatic: "dacă se află pe piaţă că noi realizăm aici o specializare bună, atunci important nu va fi dacă femeia a studiat la o şcoală pentru femei, ci cît de capabilă este".

(continuare în pag. 9)

(COMPUTER LIVE 12/90, Veronika Mirschel)

A noua, vă rog !

Simfonia a noua a lui Ludwig van Beethoven, în cuvinte şi imagini, cu sunet digital Hi-Fi : noul eveniment multi-media pentru Mac.

Sloganul actual în branşa calculatoarelor este multi-media. Prin această denumire seacă se înţelege îmbinarea tuturor mediilor existente, ca: vorbire, muzică, imagine şi video, într-un supermediu. Tehnologia multimedia se foloseşte în momentul de faţă mai ales în domeniul şcolarizării. "Elevii" cutreieră cu success munţi întregi de ştiinţă prin metoda do-it-yourself (fă singur) în mod interactiv, adică în dialog cu ecranul. Cu ajutorul tastaturii sau a unei zone sensibile la atingere, reprezentată printr-un buton pe ecran, ei îşi conduc singuri "excursia de date".

Acest mod de transmitere a cunoştiinţelor nu numai că face mai multă plăcere, dar este şi mult mai eficient. Nu doar pentru că învăţînd astfel cunoştiiţele se fixează mai bine, ci şi pentru că utilizatorul unui astfel de sistem îşi poate fixa singur ritmul de învăţare. Nu din întîmplare această metodă este introdusă deja de o serie de intreprinderi în cadrul şcolarizării personalului cu ajutorul calculatorului (Computer Based Training CBT).

Important pentru înţelegerea sensului şi construcţiei unei astfel de producţii multi-media în scopuri de CBT este faptul că un astfel de sistem are avantajul de a putea oricînd rechema informaţii despre fiecare noţiune superioară. Multimedia este pe cale de a se dezvolta tot mai mult şi în domeniul calculatoarelor. Pînă în prezent au fost arătate în public doar superproducţii, ticsite cu video, animaţie, muzică, zgomote şi vorbire, probabil pentru a etala toate posibilităţile. O astfel de furtună de informaţii scurte cere o dotare costisitoare hard. Memoria principală a calculatorului trebuie să fie de mai mulţi MBytes, şi de cele mai multe ori sînt necesare şi hard-discuri de mare capacitate (mai multe sute de MBytes). Foarte des sînt folosite picupuri de imagini (răspîndite mai ales în SUA ca playere pentru discuri video).

Calculatorul multi-media cel mai bun în acest moment este considerat a fi Apple Macintosh, datorită posibilităţilor de grafică şi sunet excelente. Aici, în lumea Macintosh, a apărut o nouă tendinţă, cea a "întoarcerii la simplitate".

Exemplificăm această tendinţă prin producţia multi-media "Ludwig van Beethoven, simfonia a 9-a". Pentru a ne familiariza cu viaţa şi opera lui Beethoven pe această cale, nu avem nevoie decît de un Macintosh şi o unitate CD-ROM (pentru memorarea datelor, textelor şi a graficii, cu o capacitate de aproximativ 650 MByte). În ceea ce priveşte calculatorul, ne ajunge chiar şi modelul cel mai "mic", noul Macintosh Classic, sau referitor la tipurile mai vechi, Macintosh SE. Mac Classic, prezentat în octombrie, costă aproximativ 3000 mărci.

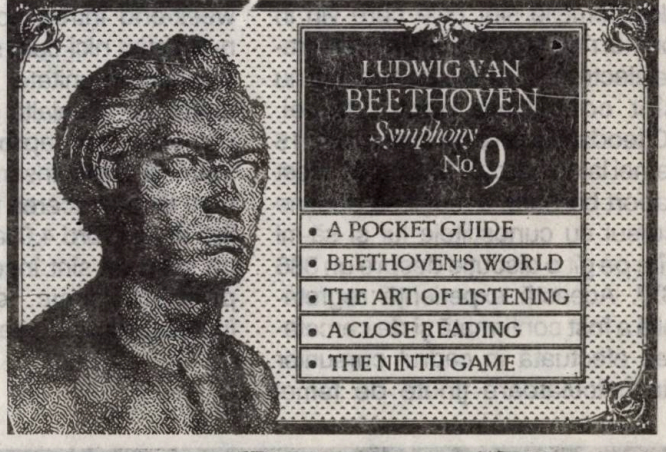
Unităţi CD-ROM se pot obţine pentru circa 1500 mărci. Ele au avantajul că de pe ele, cuplate cu un Macintosh, se pot prelucra atît discuri compacte CD-ROM cu date, texte, imagini şi grafică, şi de pe aceste unităţi se pot asculta şi discurile audio compacte uzuale după dorinţă: prin căşti sau prin instalaţia stereo.

Referitor la simfonia a 9-a a lui Beethoven, se poate cumpăra un disc compact cu simfonia a 9-a cît şi două dischete cu baza de date propriu-zisă, la un preţ de circa 400 mărci. CD-ul este o înregistrare comercială a simfoniei a 9-a. Cine se satură de învăţat poate să se delecteze ascultînd muzică.

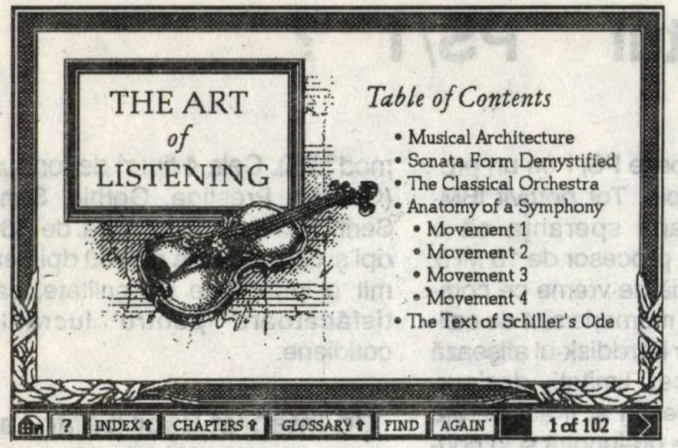
Baza de date se bazează pe programul multi-media "Hypercard", ea conţine texte voluminoase, grafice şi imagini ilustrate. Prin simbolurile de pe ecran se poate accesa discul compact din unitate. Astfel se pot derula anumite pasaje din simfonia a 9-a, bineînţeles stereo şi de calitate Hi-Fi. Baza de date este împărţită în 5 părţi, legate între ele. Meniul scurt permite accesarea directă şi ascultarea diferitelor pasaje din simfonie.

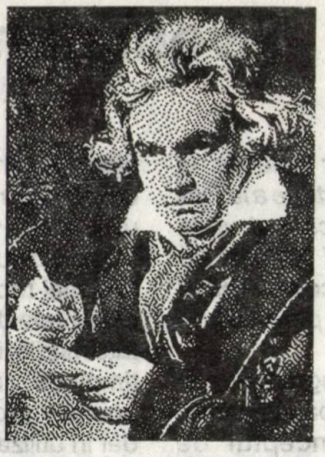
Pe 124 de pagini ilustrate capitolul "Beethoven's World" ne poartă în lumea marelui maestru. Obţinem informaţii atît din punct de vedere istoric cît şi despre problemele şi tragediile lui personale. Se subliniază atît influenţa lui Schiller asupra lucrărilor lui Beethoven cît şi importanţa celor doi în spiritul acelor timpuri. Prin apăsarea unei taste putem răsfoi însemnări interesante din jurnalul lui zilnic sau să ascultăm secvenţe din schiţe muzicale ale lui Beethoven. Practica concertistică de atunci şi de acum sînt comparate în detaliu şi ni se înşiră bibliografia pentru aprofundarea materiei.

Introducerea în "Arta auzului" ex-



Meniul principal al programului





O introducere informativă în teoria muzicii clasice

plică modul în care este construită o simfonie sau ce instrumente compun o orchestră şi care sînt caracteristicile lor. Textele sînt relaxate prin scanarea în cadrul textului a unor imagini contemporane, pasaje de muzică din CD şi prin melodii produse direct de Macintosh. Plictiseala nu poate să apară.

Esenţa propriu-zisă o constituie partea "A close reading" (o citire atentă), care reprezintă o consideraţie aproape ştiinţifică muzicală asupra simfoniei a 9-a. Textele explică de ce Beethoven a compus respectivele pasaje aşa şi nu altfel. În plus se explică şi termeni muzicali de specialitate.

După dorinţă se pot vedea notele pasajelor alese care în timpul derulării respectivele pasaje vor ieşi în evidenţă. În această parte a sistemului utilizatorul are loc pentru notiţe proprii. Acest produs se dovedeşte a fi o formă cu adevărat nouă a transmiterii cunoştiinţelor, printr-un fel de "control al scopului învăţării", ce se face foarte lejer sub forma unui joc întrebare-răspuns: un fel de "cine ştie, cîştigă". Este vorba de a răspunde la întrebări de dificultate diferită din 3 categorii de probleme (exemple de ascultare, instrumentaţie, viaţa şi opera). Chiar maestrul Beethoven în persoană apare ca prezentator, cicăleşte, laudă şi face cu ochiul, în funcţie de necesitate.

Modul acesta de îndeletnicire cu simfonia a 9-a a lui Beethoven nu este interesant doar pentru muzicologi sau pentru cei ce doresc să devină muzicologi.Acest produs multi-media este recomandat chiar şi persoanelor care nu-şi doresc să strălucească neapărat în concursuri "cine ştie, cîştigă". Deoarece face o imensă plăcere să înveţi să înţelegi un artist prin joc, el este cadoul ideal pentru posesorii de Mac iubitori de muzică.

Această parcurgere Beethoven este prima dintr-o serie întreagă, care vor apărea în curînd în comerţ, produse de firma americană Warner New Media. Produse asemănătoare despre Flautul fermecat al lui Mozart şi Igor Stravinsky sînt deja terminate. Singurul handicap al acestor pachete informaţionale muzicale : textele sînt disponibile, din păcate încă, doar în limba engleză.

Cu aceste produse noi este posibil un salt de calitate aproape fascinant al aplicaţiilor pe calculator, atît în şcoli, universităţi, biblioteci, cît şi acasă. Concluzia: jocurile pe calculator şi televizorul sînt depăşite, multi-media este în vogă.

(COMPUTER LIVE 12/90, Stefan Horst)

Destinse fără frică şi vesele

(continuare din pag. 7)

Cursurile se bazează, conform programei şcolii de calculatoare pentru femei, pe deviza: "nu mai mult decît trebuie, nu mai puţin decît este necesar!" Cursantele confirmă acest motto: "cred că ştiu încă tot ce am învăţat în acest week -end, fără ca seara să mai trebuiască să studiez".Pentru a contracara ideea preconceputâ despre femeile de la bucătărie, încă din anul '80 guvernul a sprijinit proiecte model, în care femeile vor fi instruite de femei. "Dacă la început lucrul pe calculator a fost impregnat de teamă, lipsă de independenţă, şi nesiguranţă, cu timpul mînuirea calculatorului a devenit mai sigură, mai independentă şi mai evidentă, cum nici o femeie nu şi-ar fi putut închipui la început", se spune în raportul proiectului "Deutscher Frauenring e.V". Aceste rezultate ale cercetării se suprapun experienţei din practică pe care docentele cursurilor pentru femei de la Universitatea din München le-au putut face. La femei întrebarea "ce face calculatorul cu mine, cu gîndurile mele?" iese mai tare în evidenţă decît la bărbaţi.

De aceea cursurile pentru femei oferă şi timp pentru discuţii, de exemplu, despre problema "prelucrarea electronică feminină a datelor, există aşa ceva?". La etajul clădirii vechi mai există încă două încăperi. Atunci cînd ele vor fi reamenajate şi aici se vor ţine cursuri. O practicantă de medicină face cursuri de masaj - o apropiere feminină de tehnologia calculatoarelor este mai mult decît o simplă discuţie profesională despre hardware.

Evenimentul PS/1 ?

Un microcalculator puternic la îndemîna tuturor. O victorie a la Pyrrhus pentru Apple, care-şi vede strategia convivialităţii preluată de IBM. Şi totuşi, ce aduce nou PS/1? "Lucrări" pe un sistem personal.

Era utilizatorilor programatori s-a încheiat odată cu apariţia primului Macintosh. Conceptul de calculator pentru un utilizator neavizat a penetrat greu în lumea PC-urilor, dar s-a impus chiar şi în faţa lui "Big Blue" care se vede nevoit să-şi abandoneze politica de subordonare a utilizatorului faţă de specialist. Interfeţele utilizator iconografice cu suport mouse se dezvoltă în ciuda coexistenţei cu un sistem de operare în mod linie de comandă - MS-DOS - şi a varietăţii de standarde grafice hard şi soft. Şi iată că, mai bine mai tîrziu decît niciodată, IBM propune un concept care nu mai e original, dar rămîne foarte actual: un sistem personal accesibil oricui.

PS/1 - o nouă generaţie ? Este noul venit atît de diferit pentru a impune un nou standard? Rămîne de văzut. Din punct de vedere al performanţelor nu este altceva decît un AT ca toate celelalte. Procesorul 80286/10MHz îl prezintă ca pe o maşină modestă, fără nimic extravagant. Dincolo de aparenţele fişei tehnice vom (re)descoperi calităţile unui calculator compact: dimensiuni foarte reduse, grad mare de integrare, opţiuni uşor instalabile şi nu în ultimul rînd un preţ accesibil la o calitate IBM. Iar pentru ca similitudinea cu Macintosh să fie completă, interfaţa seacă a MS-DOS este rapid înlocuită după boot-are cu imaginea mai prietenoasă a integratului MS Works. Utilizatorul neavizat poate demara rapid cu ajutorul mouse-ului un ciclu de autoinstruire sau chiar o aplicaţie.

Calitatea înainte de toate - constructorul nu a cedat tentaţiei care animă majoritatea concurenţilor. IBM afirmă că nimic nu l-ar fi împiedicat să echipeze PS/1 cu un procesor mai rapid. Tot potrivit IBM-ului, este vană speranţa că o maşină cu un procesor de 16 MHz ar fi mai rapidă de vreme ce componentele de memorie sînt de calitate medie, iar harddisk-ul afişează timpi de acces limitaţi; desigur, anumite procesări ar fi accelerate dar în utilizare prelungită şi în condiţii normale de lucru dezechilibrul sistemului determină o uzură precoce iar riscurile de pană cresc. Oricum, decizia unui ansamblu echilibrat de componente este semnificativă pentru calitatea materialului. Şi altceva? Salutară este prezenţa adaptorului VGA integrat pe motherboard. Lizibilitatea informaţiei pe ecran este mult ameliorată astfel, fapt perfect normal pentru un calculator care se doreşte convivial. Şi pentru că IBM nu face compromisuri, clientul are de ales doar între un monitor color şi unul monocrom, standardul grafic rămînînd acelaşi. Harddisk-ul de 30 Mbytes şi timp de acces mediu de 21 ms se pretează la numeroase aplicaţii exigente. Şi aici, clientul cu posibilităţi modeste sau ale cărui necesităţi nu depăşesc cadrul lui Works, poate opta pentru o a doua unitate de 3,5"/1,44 Mbytes în locul harddisk-ului. Mouse-ul, compatibil Microsoft, este sobru dar agreabil - de fapt este acelaşi care echipează familia PS/2; cablul foarte lung permite şi de la distanţă o utilizare lejeră a sistemului. O menţiune specială pentru livrarea în configuraţia standard a modemului V23 "Light Modem" al societăţii Com 1. De dimensiunile unui pachet de ţigări, se cuplează la ieşirea serială şi la reţeaua telefonică. Modem-ul este livrat în corespondenţă cu programul de comunicaţie Overcom al societăţii Goto Informatique.

Concomitent cu PS/1, IBM prezintă imprimanta matricială 2205 model 1. De tip 24 ace, bidirecţional, format A4, viteza ei de imprimare este de 192 cps în mod draft şi 53 în mod NLQ. Cele 4 tipuri de font-uri (Courier, Prestige, Gothic, Sans Serif), rezoluţia orizontală de 360 dpi şi cea verticală de 180 dpi permit o imprimare de calitate, satisfăcătoare pentru lucrările cotidiene.

Un integrat uşor pentru a porni: Works 2

Un integrat este un remarcabil instrument de iniţiere în microinformatică, iar Works îşi demonstrează din plin această calitate. Odată cu procesorul de texte se învaţă mai bine de 50% din tehnica ... lucrărilor. Structura meniurilor şi opţiunile odată asimilate, nu mai rămîne de înţeles decît finalitatea diferitelor module (calcul tabelar, gestiunea bazelor de date, etc.) Remarcabilă este şi facilitatea de schimb de date între diferite module, chiar dacă nu de o manieră DDE (Dynamic Data Exchange); spre exemplu un raport conceput cu procesorul de texte care include şi o parte dintr-un tabel realizat sub un program de calcul tabelar va fi actualizat ori de cîte ori tabelul este modificat dacă în prealabil se definesc legături DDE.

Tehnica Windows: opt ferestre conţinînd fiecare un fişier pot fi afişate simultan pe ecran. Works integrează de asemenea şi o reprezentare iconografică, iar utilizatorul are de ales între două moduri de lucru: text (25 sau 50 de linii) şi grafic (60 de linii în mod VGA). Este posibilă astfel opţiunea între viteză şi convivialitate, prima lăsînd deseori de dorit sub Windows.

Trusa de scule: nici un accesoriu de birou nu a fost neglijat: calculator, ceas şi calendar sînt disponibile în orice modul. Funcţiile DOS sînt binevenite pentru orice neavizat care ignoră sintaxa comenzilor corespunzătoare. Remarcabilă rămîne posibilitatea "point and shoot" de a lansa orice alt program de sub Works.

Un **procesor de texte** dichisit: fără a aspira la puterea lui Word, procesorul de texte este bine conceput şi remarcabil prin simplitate. Dacă este de neimaginat compunerea unei lucrări de 500 de pagini sub Works, el se pretează totuşi foarte bine la redactarea unei corespondenţe cotidiene. În afara funcţiilor uzuale de editare, un foarte complet dicţionar precum şi dicţionarul de sinonime sînt plăcute surprize (evident, pentru utilizatori anglofoni, francofoni,...). Din păcate editarea nu se face în mod WYSIWYG (what you see is what you get) (care "garantează" că ceea ce va ieşi la imprimantă este identic cu ceea ce se vede pe ecran), hap greu de înghiţit în mod grafic şi mal ales VGA. Pentru a compensa lipsa, Works propune o previzualizare înaintea imprimării, cam strîmtă şi greu lizibilă în absenţa unei funcţii de zoom.

Tabele şi grafice: capacitatea modulului de calcul tabelar este de 4096 linii pe 256 coloane. Deşi selectarea celulelor cu mouse-ul este foarte facilă, autorii, precauţi, au prevăzut şi un sistem de numerotare stil Multiplan. Departe de posibilităţile unor Excel sau Quattro Pro, Works propune numeroase funcţii matematice, statistice sau financiare, acceptabile pentru o utilizare curentă: urmăriri de bugete, calcule previzionale. Fără a pretinde puneri sofisticate în pagină, s-a dovedit imposibilă selectarea unui anumit tip de caracter, cu toate că şi această opţiune figurează în meniu. Pentru a reuşi aceasta, trebuie exportat tabelul în procesorul de texte, operaţiune relativ simplă de altfel. Menţinîndu-se în limitele 2D, modulul grafic are variate opţiuni de reprezentare: histograme, curbe, grafice sectoriale, densităţi de puncte, etc. Modificarea legendelor, titlurilor permit personalizarea graficelor, încercarea de a manipula însă graficele se dovedeşte o întreprindere temerară, în contradicţie cu pretenţiile de soft didactic. În sfîrşit Works nu permite imprimarea unui tabel şi a unui grafic pe aceeaşi pagină.

Gestiune de baze de date: în intenţia autorilor acest modul îşi propune familiarizare începătorului cu noţiunile de înregistrare, cîmp, raport, bază de date. Departe însă de a răspunde rigorii conceptuale în proiectarea bazelor de date, Works 2 rămîne cu o singură pretenţie: didacticismul. Fără a intra în detalii, o bază de date este în mod obligatoriu multifişier în timp ce Works este monofişier. Ar fi fost de dorit ca Microsoft să fie mai reţinut în publicitatea făcută acestui modul simplisim şi net depăşit de managerul de baze de date al unor integrate de acelaşi calibru precum First Choice şi Open Access Junior.

Comunicaţii: expedierea sau recepţia de fişiere la distanţă - acesta este rolul acestui modul; la aceasta se adaugă şi posibilitatea de a emula un terminal. Posesorii de PS/1 au la îndemînnă un obiect dedicat: Overcom.

Works rămîne un soft omogen şi face un bun mariaj cu PS/1. Fără funcţionalităţi excepţionale dar fără defecte insuportabile - iată caracterizarea acestui integrat de generaţie nouă. Fără a privilegia un modul în maniera integratelor grele - Open Access sau Symphony - Works prezintă un excelent echilibru între diversele componente - un atu important pentru vocaţia sa iniţiatică.

Overcom sau cum să scapi de singurătate

Livrarea lui PS/1 în Europa a început prin cea mai puternică reprezentanţă "Big Blue": IBM France. Pentru a se integra specificului local - reţeaua de telecomunicaţii Minitel - IBM a optat pentru produsul societăţii Goto Informatique: Overcom. Prezentarea unui emulator complet al modului de lucru Minitel nu se face în defavoarea caracteristicilor de bază ale unui program de comunicaţii care se respectă. Overcom nu se adresează exclusiv francezilor. Modul memorare permite 'învăţarea' tuturor acţiunilor utilizatorului. În timpul accesului la un server o anumită secvenţă de răspunsuri poate fi "învăţată" de Overcom, facilitate deosebită în consultarea unui cont bancar, căutarea unui orar de curse aeriene sau accesarea unor baze de date profesionale. "Internaţionalismul" transpare şi prin recunoaşterea celor mai uzuale protocoale de comunicţie, între care X/Y Modem, Kermit şi evident Goto. Descoperirea unui editor de texte şi a unui limbaj de programare indică existenţa unui nivel superior de utilizare a modului meniu - care răspunde problemelor cotidiene dar nu corespunde exigenţelor unui transfer complex şi important de date, cu conexiuni la anumite ore şi însoţit de prelucrări. Interpretorul de comenzi permite atît lucrul în mod linie cît şi execuţia de fişiere de comenzi. Remarcabilă este posibilitatea de a pune la punct o aplicaţie sub controlul unui depanator (debugger) intern care permite urmărirea execuţiei fiecărei comenzi şi localizarea sub editorul de text a liniei incorecte.
Limbajul de programare conţine peste 100 de comenzi, grupate în opt clase: comenzi de gestiune a comunicaţiilor, a funcţiilor video, de tastatură, de acces la disc, comenzi de supraveghere a liniei şi tratare a şirurilor de caractere în paralel cu procesarea comenzilor directe, comenzi de configurare Overcom, funcţii videotex (specifice terminalului Minitel) precum şi instrucţiuni condiţionale. O menţiune aparte pentru existenţa unor tipuri variate de variabile şi a posibilităţii de grupare a comenzilor în proceduri.

Overcom se conturează ca un program profesional în contextul exigenţelor de convivialitate al "filozofiei PS/1". Cu toate acestea el rămîne foarte abordabil în modul de lucru cu meniuri, iar realizatorii lui fac eforturi spre a-l impune şi neprofesioniştilor anunţînd de curînd versiunea operabilă cu ajutorul mouse-ului.

(ing. Alin Flaidăr)

Unix şi viruşii

Sistemele multiuser, respectiv sistemele de calcul înglobate în reţele, sînt foarte expuse înspre afară. În mod deosebit calculatoarele Unix, racordate în multiple feluri la reţeaua mondială, sînt în mod deosebit periclitate.

Recent, prin condamnarea studentului american Robert T. Morris, s-a încheiat un caz care a generat enorm de multe discuţii relativ la securitatea datelor şi în mod deosebit a sistemelor Unix.

Robert Morris este cel care a creat acel virus care în noiembrie 1988 a infectat reţeaua americană Internet. Această reţea acoperă întregul continent american şi leagă practic toate sistemele Unix mari din S.U.A. Există şi o legătură înspre reţelele militare. Deşi virusul în mod evident nu a fost gîndit să producă pagube în sistemele atacate, datorită blocajelor de sistem şi utilizatorilor speriaţi care s-au deconectat de la reţea s-au produs pagube cifrate la cîteva milioane de dolari. Părea demonstrată în felul acesta sensibilitatea deosebită a sistemelor Unix la astfel de atacuri, prezisă de experţi. Dar oare sînt sistemele Unix într-adevăr foarte sensibile?

Pentru a răspunde la această întrebare, merită făcută o inventariere a diferitelor surse de pericol care ar putea ameninţa securitatea calculatoarelor. Principial, tema are două aspecte: pe de o parte securitatea vizavi de pierderi de date prin erori de sistem, pe de altă parte protecţia faţă de accesul neautorizat şi distrugere. Ne vom ocupa în continuarea articolului numai de cel de-al doilea aspect al problemei.

Un caz limită este distrugerea datelor de către utilizatori autorizaţi. Fie prin utilizarea defectuoasă a unui program, din neatenţie sau pur şi simplu din neştiinţă: probabil că oricui i s-a întîmplat deja odată să distrugă date, fără să fi vrut. De aşa ceva probabil că nu e sigur nici un sistem. Dar desigur că există caracteristici de sistem care fac mai mult sau mai puţin probabile astfel de incidente. Că cele mai multe comenzi Unix au destul de multă încredere în utilizator şi execută şi comenzi distructive fără să se mai asigure prin întrebări suplimentare poate fi cotat din acest punct de vedere ca destul de periculos. Şi cuprinzătoarele posibilităţi disponibile prin Unix-Shell pot avea efecte neplăcute în caz de programare sau utilizare defectuoasă. Deşi aici se pot aduce critici serioase Unix-ului, per total gradul său de securitate faţă de erorile utilizatorului este mai mare decît cel de sub MS-DOS. Distrugerea accidentală a datelor sistemului sau ale altor utilizatori este practic exclusă sub Unix, pe cînd un utilizator MS-DOS nepriceput, în ziua lui bună, poate face sistemul inutilizabil fără să se obosească prea mult.

În discuţia generală asupra securităţii, pericolul reprezentat de utilizatorii autorizaţi este lăsat de-o parte. Punctul de interes al discuţiei este securitatea faţă de accesul neautorizat în sistem. Astfel de intruşi pot fi atît persoane cît şi programe. Urmările pot varia de la neplăceri mai mult sau mai puţin grave, ca în cazul virusului Internet, pînă la furtul sau distrugerea unor date cu daune în valoare de milioane. Pericolul unor intruşi din afară, a aşanumiţilor "hackeri", este estimat în general a fi cel mai mare. Există o contramăsură aproape sigură şi anume controlul absolut al accesului fizic la sistem. Dacă toate terminalele şi calculatorul se află în încăperi la care accesul este complet controlat, nu poate lucra nimeni neautorizat la sistem. În practică însă, exceptînd poate domeniul militar, astfel de soluţii nu pot fi luate în considerare. Chiar şi în întreprinderi care îşi pot permite controlul necesar, protecţia obţinută nu este fără fisuri.

La considerarea accesului la echipamentele de calcul, nu trebuie uitat subiectul reţea. Interconectările între diverse sisteme cresc mereu pe plan mondial, iar la sistemele Unix racordarea la reţeaua mondială este o chestiune aproape subînţeleasă. Posibilităţile pentru o imixtiune din afară cresc desigur mult prin aceasta. Că îşi crează acces persoane, prin dialog la terminal, sau programe corespunzătoare, este lipsit de însemnătate în acest context.

Mai ales atunci cînd legătura cu reţeaua este intens folosită, există interes pentru un acces confortabil la reţea şi la alte calculatoare. Astfel încît puţini vor fi utilizatorii care vor fi dispuşi să introducă mai multe parole, numai pentru a putea comunica cu un alt calculator. Dacă însă acest lucru nu se întîmplă, există pericolul ca intrusul să se poată răspîndi în toată reţeaua, după ce a reuşit să se infiltreze într-unul din sistemele conectate. Printre altele, răspîndirea rapidă a virusului Internet a fost posibilă şi pentu că între multe sisteme din reţea există o 'înţelegere' relativă la marcaje "de încredere", la care nu se mai face şi a doua verificare a parolei. Din nefericire şi utilizarea reţelei pentru schimbul de ştiri via poştă electronică este un punct slab al securităţii datelor din acest punct de vedere. Prin exploatarea unor greşeli în programele respective, virusul Internet s-a putut multiplica folosind chiar acest mecanism.

Ce este pentru Unix reţeaua, este pentru multe alte sisteme reţeaua telefonică. Cel tîrziu de la filmul "War Games" (jocuri de-a războiul) fiecărui utilizator ar trebui să-i fie clare pericolele ce pîndesc un sistem în care se poate intra via

telefon. Cel puţin în ce priveşte acest aspect, filmul este foarte aproape de realitate. Pentru astfel de instalaţii, protecţia simplă a marcajelor prin parole nu ajunge în toate cazurile. În astfel de situaţii se obişnuiesc măsuri de protecţie mult mai cuprinzătoare. Astfel de exemplu este uzual ca după mai multe încercări nereuşite de Login, să se întrerupă legătura. Dar hackeri perseverenţi nu se vor lăsa descurajaţi de această măsură. O înregistrare a unor astfel de încercări ar trebui să fie regula, ca de altfel şi controlul activităţilor desfăşurate prin legătură externă, mai ales noaptea. Cea mai simplă şi mai bună metodă este deseori deconectarea telefonului.

Între "atacurile" efectuate de persoane şi cele efectuate de programe există deosebiri principiale. Programele sînt numite de regulă "viruşi". Există o varietate de subspecii, printre ele "caii troieni", care spre exterior se prezintă ca utilitare folositoare. Apoi "bombele cu ceas" care "explodează" abia la un moment bine determinat.

Spre deosebire de modelul din antichitate, caii troieni pe care îi discutăm aici necesită în cele mai multe cazuri existenţa unei posibilităţi de acces la sistem. Scopul artificiului este de a obţine drepturi mai mari în sistem. Sub Unix, caii troieni pot profita de faptul că în anumite condiţii programele pot avea alte drepturi decît utilizatorii care le-au lansat. Pentru programe sistem "adevărate", această proprietate nu numai că are sens, da nu poate fi evitată. Un exemplu bun este programul pentru modificarea parolei. Sigur că utilizatorul obişnuit nu are drept de scriere în fişierul de parole. Dar pentru a-şi putea modifica propria parolă, programul corespunzător trebuie să aibă acest drept. Aceasta necesită desigur o programare care să garanteze că în felul acesta utilizatorul îşi modifică numai propria parolă. Un factor de risc devine această proprietate atunci cînd aşazisul program de sistem este în realitate un virus, care în afara funcţiei propriu-zise mai urmăreşte scopuri ascunse, rareori curate. Dacă unui astfel de program i-a reuşit odată să-şi creeze drepturi sporite în sistem, soarta acestuia este cel mai adesea pecetluită. Un exemplu deosebit de "mîrşav" pentru specia cailor troieni sînt falsele Login-Shell-uri. Ele se substituie Login-ului din sistem şi se prezintă utilizatorului ca de obicei, determinîndu-l să-şi dea parola şi marcajul. Dacă utilizatorul astfel indus în eroare îşi dă datele, nu i se face încă intrarea în sistem ci e păcălit în continuare printr-um mesaj de eroare. Login-ul "fals" ţine minte parola şi iese din scenă, lăsînd Loginul adevărat să-şi facă treaba, astfel că cel mai ades următoarea încercare de Login reuşeşte fără probleme. Cum oricui i se mai întîmplă să tasteze greşit vreun caracter din parolă, procedura nu trezeşte nici un fel de suspiciuni. Cu puţin noroc, în felul acesta intrusul "pescuieşte" chiar o parolă a unui utilizator privilegiat, asfel încît pagubele ce se pot produce sistemului devin incontrolabile.

Foarte asemănător funcţionează "bombele cu ceas". Şi acestea se vor ascunde de obicei sub o mască inofensivă, fără a-şi arăta prin ceva adevărata natură. La un moment dat aceasta se dezvăluie, cel mai ades provocînd distrugerile maximal posibile. Nu rareori aceste mecanisme sînt moştenite de la foşti colaboratori, care în felul acesta vor să se răzbune pentru o prezumtivă sau reală nedreptate care li s-a făcut. Din punct de vedere al sistemului astfel de atacuri sînt aproape imposibil de prevenit, mai ales pentru că e vorba de persoane care pe baza funcţiei lor trebuie să aibă acces liber la sistem.

În lumea MS-DOS viruşii de obicei sînt programe executabile sau secvenţe de program care se răspîndesc îndeosebi prin dischete. Deoarece PC-urile sînt compatibile aproape fără excepţie chiar la nivel de cod executabil, această metodă de răspîndire este optimă. În lumea Unix acest gen de viruşi sînt mult mai puţin răspîndiţi. Hardul foarte diversificat ar face imposibilă răspîndirea largă a unui astfel de intrus. De aceea aici s-a creat un mecanism de reproducere diferit. Viruşii din Unix îşi duc de obicei sursa cu ei. Dacă virusul a reuşit să pătrundă într-un sistem, se reproduce acolo utilizînd uneltele disponibile pe acel sistem. În acest fel, el exploatează la modul optim interfaţa standardizată a Unix-ului şi este la fel de independent de hardware ca orice alt program aplicativ. Singura limitare se datorează diferitelor derivate ale Unix-ului. Astfel virusul Internet s-a putut replica numai pe calculatoarele cu Berkeley Unix, pentru că o parte din programele folosite nu există sub System V. După cum s-a văzut, aceasta nu l-a putut opri să blocheze întreaga reţea în numai cîteva ore.

Camuflajul viruşilor

În ciuda unor diferenţe semnificative, există o serie de caracteristici care se găsesc la toate programele-virus. Cea mai importantă este camuflarea. Un virus care se descoperă de la bun început nu prea are şanse de reuşită. O metodă verificată de camuflaj este "ascunderea" banală a unui program. În MS-DOS aceasta se poate face de exemplu setînd atributul de 'hidden' (ascuns) la înscrierea în fişierul director. În Unix 'ascunderea' unui fişier se poate face în cele mai simple cazuri dîndu-i un nume care să înceapă cu punct. Ambele metode sînt primitive şi expuse la descoperiri accidentale. Camuflarea stil 'cal troian' are mult mai multe şanse de reuşită. Fie sub un nume fals de utilitar sistem, fie sub aspectul unui utilitar folositor şi eficient, un astfel de virus îşi crează 'la vedere' accesul la sistem. În lumea PC-urilor, îndeosebi jocurile sînt posibili inamici mascaţi.

Pasageri clandestini prin sisteme

O altă caracteristică a mai tuturor virușilor este tendința de răspîndire și reproducere. Rareori un astfel de intrus se mulțumește cu a fi infestat un singur sistem. Răspîndirea molimei se poate face prin cele mai diferite trucuri. O regulă de bază este de a nu te face repede remarcat. Un sistem recunoscut ca infestat micșorează drastic șansele virusului de a se mai răspîndi de pe el.

Atît pentru camuflare cît și pentru răspîndire este deseori folosită tehnica 'agățării' de alte programe, nevinovate, pentru a începe astfel călătoria spre alte sisteme ca un pasager clandestin. Varianta este destul de neplăcută, pentru că nici cunoașterea purtătorului original nu-i mai ajută utilizatorului.

Din fericire, nu în toate cazurile virușii au tendința să distrugă datele. În cazuri limită însă, sisteme complete devin nefolosibile, ceea ce denotă însă și insuficiente măsuri de protecție. S-au aflat ocazional cîteva erori din Unix care au făcut posibile astfel de incidente, însă aceste 'găuri' s-au peticit totdeauna repede după ce au ieșit la iveală. În antiteză, la un sistem lipsit de protecție cum este MS-DOS, chiar și distrugerea unor date de sistem e greu de împiedicat după ce un virus a pătruns.

Se pot găsi o multitudine de contramăsuri împotriva virușilor în diversele lor variante. Au devenit populare așa-numitele 'Virus-checker'. Deseori acestea se bazează pe principiul că dimensiunea și eventualele caracteristici ale programelor ce trebuiesc protejate sînt cunoscute, astfel că orice modificare determină o alarmare. Asta funcționează cu succes la clasa de viruși 'agățători' și desigur poate fi eficient numai dacă activarea programului s-a putut face cît timp sistemul era "sănătos". Alte programe de protecție cunosc "tiparul" mai multor viruși și-i caută pe aceștia. Această metodă are sens la sistemele infestate. E nevoie însă de actualizări rapide ale programului de protecție în momentul apariției unui virus nou. Autorii virușilor vor avea totdeauna un ușor avantaj din acest punct de vedere.

Așa se recunosc virușii

Și un control 'manual' poate fi de folos. Cîte un virus se deconspiră prin activități care pe un observator atent îl pot preveni din timp. Astfel de 'semne' sînt de exemplu timpi de încărcare neobișnuit de lungi sau, ceea ce mai ales la PC-uri se observă ușor, accese la disc la momente de timp neașteptate. Virusul Internet de exemplu s-a făcut remarcat prin fișiere inexplicabile ce apăreau într-un director prevăzut pentru fișiere temporare.

O anumită protecție o pot oferi și sisteme corespunzătoare orientate spre securizarea datelor. Atît penetrarea cît și răspîndirea sînt vizibil îngreunate. La o organizare modulară corespunzătoare a sistemului cresc șansele de a păstra eventualele pierderi în limite rezonabile.

Cel mai sigur mijloc de prevenire, ca la toate molipsirile, este 'igiena'. În cazul protecției datelor aceasta înseamnă mai ales: atenție la orice schimb de date. Cine nu are neapărată nevoie de instalații potențial riscante ca rețelele de calculatoare sau legăturile telefonice, face mai bine renunțînd la ele.

Dincolo de acestea există desigur și măsuri tehnice de protecție ca cele prevăzute în Orange Book. Desigur însă nici o încadrare superioară după 'cartea portocalie' nu este neapărat o garanție pentu un standard global de securitate ridicat. Și aici intră în joc factorul "utilizator". Căci și cele mai strălucite mecanisme de securitate nu sînt decît atît de bune cît o permit oamenii care lucrează cu ele. Pe de o parte, prin încadrarea într-una din clasele definite în Orange Book se fac niște aserțiuni principiale despre securitatea unui sistem. Pe de altă parte, exact opusul sistemului absolut sigur este sistemul "prietenos cu utilizatorul", și aceste două obiective se exclud reciproc. Cele mai multe mecanisme ale clasificărilor superioare îngrădesc atît de mult libertatea utilizatorilor, încît periclitează acceptarea acestor măsuri. De aceea încadrarea într-unul din nivelele superioare după aceste criterii nu este de dorit fără obiecții la toate sistemele.

Mecanismul de parolă din Unix se pretează, în principiu, pentru a împiedica accesul în sistem al unor intruși. Dacă însă o mare parte a utilizatorilor folosesc acest mecanism numai superficial sau de loc, atunci sporul de securitate este foarte mic. Și acest lucru a fost demonstrat de virusul Internet. Virusul conținea o variantă rapidă a algoritmului folosit pentru cifrarea parolelor utilizator depuse în fișierul "passwd". Prin utilizarea de dicționare pe respectivele calculatoare a pornit apoi la decriptarea cuvintelor conținute în acestea, cu speranța de a "nimeri" ceva din fișierul passwd. Această tentativă simplă a ajuns la rate de reușită de pînă la 50%. Înspre sistem, trebuie criticat faptul că fișierul dicționar putea fi accesat fără opreliști în citire. Faptul că toate parolele din el erau cifrate nu este o scuză, după cum s-a văzut, în primul rînd pentru că algoritmul de cifrare este cunoscut și e relativ simplu și ușor de programat. Dar un factor esențial sînt și utilizatorii. Practica de a folosi parole simple și uzuale reprezintă o slăbire suplimentară a mecanismului de protecție prin parole. Din comoditate, foarte des se folosesc coduri foarte scurte. Ceea ce face relativ simplă "spargerea" codului prin încercări. Pe lîngă apelul către utilizatori, care în acest caz e greu de crezut că va fi un succes total, o seamă de măsuri pot fi luate la nivelul sistemului. Astfel de exemplu mecanismul de protecție prin parole poate fi determinat să nu accepte decît parole netriviale. O lungime minimă și folosirea de caractere speciale poate fi "forțată" în felul acesta. E drept că acest gen de măsuri nu au darul să stîrnea-

"Orange Book" - ce înseamnă "sigur"?

Ca multe alte "cuvinte mari", cuvîntul "siguranţă" nu spune mare lucru prin el insuşi. Fundamentele pentru posibilitatea măsurării şi comparării securităţii sistemelor de calcul sînt create abia de o mulţime de criterii verificabile. Primele încercări în această direcţie au venit, după cum era de aşteptat, din domeniul militar. La solicitarea ministerului american al apărării s-a născut încă din 1983 un studiu devenit cunoscut sub numele de "cartea portocalie" datorită coperţilor. Clasele şi criteriile stabilite acolo sînt unanim recunoscute de atunci şi nu servesc numai instituţiilor militare ca referinţă pentru achiziţionare de sisteme noi.

Cartea portocalie împarte sistemele în 4 clase principale, notate cu "D" pentru cele foarte labile pînă la "A" pentru cele extrem de sigure. În cadrul acestor grupe există o altă subdivizare în trepte. Pentru fiecare grupă există un catalog de cerinţe care trebuiesc îndeplinite pentru a permite încadrarea unui sistem în acea grupă. Fiecare treaptă conţine desigur şi toate cerinţele claselor inferioare.

În clasa D sînt cuprinse toate sistemele care nu pot fi încadrate în nici una din treptele celelalte. Astfel de sisteme nu dispun practic de nici un fel de mecanisme de securizare. în această categorie intră multe sisteme de operare de la calculatoare mari, dar şi toate sistemele de operare larg răspîndite pentru PC-uri, cum ar fi MS-DOS.

Clasa C se divide în grupele C1 şi C2. Pentru încadrarea în C1 o premisă este existenţa unui mecanism de protecţie la acces. Orice utilizator are anumite drepturi de acces la date, care pot fi şi transmise. E prevăzută şi acordarea de drepturi unor grupe de utilizatori. Prin aceasta sînt create premisele de bază ca tuturor utilizatorilor să le fie accesibile datele de care au nevoie şi nimic în plus. C2 cere o structurare mai fină. Trebuie să se poată stabili şi la nivel de fişier care utilizatori şi în ce mod anume au acces. Sînt necesare şi mecanisme de protocol pentru supravegherea acţiunilor întreprinse de utilizatori.

Mecanismele de acces ale Unixului pledează pentru încadrarea în clasa C1. Unele probleme mici şi care principial pot fi rezolvate uşor ar face însă discutabilă această încadrare dacă s-ar face o examinare formală. Astfel fişierul "passwd" în care sînt depuse parolele cifrate poate fi citit de toţi utilizatorii. Acest lucru contravine cerinţelor C1 şi poate însemna un factor de risc real în practică.

Aceasta şi încă cîteva probleme au fost eradicate de o serie întreagă de ofertanţi de Unix. O încadrare în C2 nu poate fi realizată pentru că pentru diferenţierea drepturilor de acces nu există decît trei grupe de utilizatori: proprietarul, grupa proprietarului şi toţi ceilalţi. O acordare realmente individualizată de drepturi de acces este imposibilă. După cum o demonstrează însă diverse sisteme bazate pe Unix, acest obstacol poate fi trecut. în zona PC-urilor trebuie menţionat aici noul SCO-Unix System V/386 Release 3.2, care satisface aceste cerinţe, păstrînd compatibilitatea datelor către versiunile mai vechi.

Clasa B se divide în 3 subgrupe. B1 cere mai întîi un control obligatoriu al accesului. B2 necesită o arhitectură modulară a sistemului. Asupra cerinţelor şi mai speciale din B3 şi A nu se va mai insista aici, deoarece cerinţelor formulate de acestea nu li se poate da curs decît cu eforturi deosebite, ceea ce face ca ele să nu prea aibă importanţă în practică. în prezent este cunoscut un singur sistem care a putut fi încadrat în grupa A.

Deşi deja treapta B1 poate fi atinsă numai cu mari eforturi de Unix, există cîteva încercări în această direcţie. Unii producători oferă şi "seturi de completare" cu ajutorul cărora Unix-ul poate fi ridicat pînă la nivelul B1. În ultimul timp unii candidaţi Unix se prezintă la examenul B2; printre ele şi noul System V.4 de la AT&T. O restructurare semnificativă a kernel-ului Unix este în orice caz o premisă.

Se pune şi întrebarea dacă orice calculator trebuie să îndeplinească cerinţe de securitate complexe. Pentru utilizatori care nu lucrează cu date deosebit de pretenţioase, necesitatea unui standard de securitate ridicat este discutabilă. Apoi, toată afacerea mai este şi o problemă de preţ, căci securitatea ridicată se plăteşte cu un preţ corespunzător.

Punctul de greutate pentru utilizatorul "obişnuit" îl vor reprezenta în viitor probabil treptele C2 şi B1. Aici se realizează un compromis între siguranţă şi utilizabilitate în viaţa de zi cu zi. Pentru utilizatorul mai sensibil intră în discuţie şi B2, pe cînd B3 şi A nu par potrivite decît în cazuri excepţionale.

scă entuziasmul utilizatorilor, însă securitatea sistemului are de cîştigat în mod sigur. Împreună cu un mecanism de control care urmăreşte încercările nereuşite de log-are şi întrerupe legătura, anunţînd totodată şi administratorul sistemului, pot fi evitate măcar încercările mai simple de a pătrunde în sistem din afară.

(continuare în pag.28)

(PC+Technik 12/90, Stefan Wende)

Imprimanta - sau fără hîrtie nu se poate

Vă prezentăm succint principalele tipuri de imprimante folosite pe lîngă calculatoarele personale, creionînd în fiecare caz avantajele şi dezavantajele principale. Articolul se încheie cu un tabel de sinteză, menit a prezenta cele mai cunoscute imprimante din fiecare clasă.

Imprimanta cu ace

Majoritatea utilizatorilor lucrează cu imprimanta cu ace, deoarece aceasta reprezintă alegerea cea mai ieftină (între 400-2500 DM).

Cele mai uzuale sînt imprimantele cu 9 sau 24 ace, dar există şi modele "exotice" cu 18 sau 48 de ace.

Principiul de funcţionare: scrisul şi grafica sînt realizate pe hîrtie, prin impactul produs de ace din metal preţios, acţionate de electromagneţi, peste o bandă tuşată, asupra hîrtiei. Acele, în funcţie de tipul imprimantei, sînt plasate pe una, două sau trei coloane.

La imprimantele cu ace se deosebesc două moduri de imprimare:

1)modul de listare rapidă (draft-modus). Caracterele şi literele sînt compuse din puncte singulare. Acest mod de lucru este recomandat în cazul listărilor mai voluminoase şi unde calitatea nu este importantă.

2) modul corespondenţă (Near Letter Quality-NLQ). Se realizează o supraimprimare de puncte alăturate, rezultînd o calitate a listării mai bună. Fireşte în acest caz viteza de tipărire este mai scăzută.

Singurul avantaj al acestor imprimante îl reprezintă costul redus.

Dezavantaje:

- sînt zgomotoase
- nu realizează o calitate de imprimare profesională

Imprimantele cu 9 ace se recomandă pentru :

- începători
- utilizatori care doresc numai imprimări rapide sau liste de date, tabele cu calitate redusă
- corespondenţă de medie calitate

Imprimantele cu 24 ace se recomandă pentru:

- imprimări rapide
- utilizatori care pentru bani puţini caută imprimante cu calităţi ridicate de imprimare, dar nu neapărat profesionale.

Imprimanta cu jet de cerneală

Deşi sînt cumpărate de către tot mai mulţi utilizatori (avînd preţuri între 1200-4000 DM) au avut pe piaţă o nemeritată poziţie de cenuşăreasă.

Imprimantele cu jet de cerneală produc caractere şi imagini prin pulverizarea fină a unor picături de cerneală pe hîrtia de imprimat. Există modele cu 12, 24, 48 de duze (cu cît mai multe cu atît calitatea este mai bună). În varianta color, culorile sînt aduse pe hîrtie din 4 patroane (fiecare conţinînd 12, 24 sau 48 de duze). Cîte un patron negru, albastru, galben şi roşu se mişcă deasupra hîrtiei pe un cărucior. Din cele 3 culori de bază imprimanta îşi mixează culorile necesare. În modul de lucru text putem lista cu patronul negru pînă la 900-1000 pagini, iar în modul de lucru grafic numărul de pagini variază între 100 şi 700 de pagini în funcţie de densitate.

Rezoluţia unei imprimante cu jet de cerneală este de 200x200 d.p.i. (dots per inch - puncte/inch).

Avantajele imprimantei cu jet de cerneală:

- costă mai puţin decît imprimantele profesionale din domeniu (cele cu laser) şi imprimă la fel de frumos şi silenţios ca şi acestea.
- imprimarea uniformă de la primul şi pînă la ultimul caracter spre deosebire de imprimantele cu ace la care uzura benzii tuşate scade calitatea imprimării.

Dezavantaje:

- sînt mai scumpe decît cele cu ace
- unele tipuri necesită o hîrtie specială pentru a obţine o calitate ridicată a imprimării
- nu se poate utiliza hîrtie în mai multe exemplare

Se recomandă pentru imprimări rapide de calitate medie.

Imprimanta termică

Ca şi imprimantele cu ace sau cu jet de cerneală, imprimanta termică face parte din familia imprimantelor matriciale.

Principiul de funcţionare se bazează pe topirea substanţei de pe banda tuşată pe hîrtie. Asemănător imprimantelor cu ace, caracterele sînt alcătuite din puncte singulare, care sînt depuse pe hîrtia de imprimat de pe banda de film prin influenţa temperaturii unei perii de imprimare asupra acesteia. Capul de imprimare trebuie schimbat periodic. (Din cauza frecventelor variaţii de temperatură). Imaginea grafică a paginii este foarte bună, aceasta obţinîndu-se la o viteză de imprimare medie.

Cheltuielile de imprimare însă sînt ridicate. Ultimele modele utilizînd un rînd de elemente termice

mai late, au o viteză de imprimare comparabilă cu cea a imprimantelor cu laser, avînd şi o rezoluţie bună: 300 dpi (puncte/inch).

Avantaje:
- foarte silenţioasă
- calitate de imprimare foarte bună

Dezavantaje:
- preţ de cost ridicat
- nu se poate utiliza hîrtie în mai multe exemplare

Imprimantele termice se recomandă utilizatorilor care doresc să obţină imagini grafice frumoase şi cu contrast ridicat.

Imprimanta laser

Experţii prevăd un viitor deosebit acestor tipuri de imprimante.Pe măsură ce scade pe piaţă numărul de imprimante cu ace vîndute, creşte numărul celor cu laser. Dacă preţurile acestora vor scădea şi în continuare (în prezent sînt între 3000-12000 DM) s-ar putea ca ele să scoată de pe piaţă chiar şi imprimantele cu 24 de ace.

Performanţele imprimantelor laser sînt mult mai ridicate decît ale celor matriciale. Pe cînd la imprimantele matriciale imprimarea se face caracter după caracter, la cele cu laser înainte de imprimare toată pagina este 'construită' în memorie, ca o imagine grafică.

Imprimantele cu laser produc litere şi grafică cu ajutorul unei raze de lumină. O rază laser sau mai diode luminiscente încarcă electrostatic un tambur de imprimare, în funcţie de caracterul de imprimat. Acest tambur se roteşte într-o baie de pulbere de toner, care aderă în locurile unde tamburul este încărcat static. Trecînd hîrtia prin dreptul tamburului, tonerul este "ars" pe hîrtie la temperatură ridicată.

Avantaje:
- relativ silenţioase
- calitate ridicată de imprimare
- sînt cele mai rapide imprimante (pînă la 12 pagini/minut faţă de 1,5 pagini/minut la imprimantele cu 24 ace)

Dezavantaje:
- cost ridicat, la care se adaugă costurile pentru întreţinere (tambur, toner)
- nu se poate utiliza hîrtia în mai multe exemplare

Se recomandă pentru:
- utilizatorii care doresc o imprimare rapidă şi de calitate ridicată, profesională, pentru texte şi grafică
- utilizatori care doresc să se orienteze spre Desktop Publishing

Consultînd cataloagele de imprimante se găsesc unele date ce trebuiesc tratate cu circumspecţie; de exemplu: în viteza de imprimare nu este inclus timpul necesar pentru pregătirea paginii de listat (care poate fi 15-20 minute!).

La achiziţionarea unei imprimante laser numai anumite teste speciale de imprimare ne pot lămuri asupra performanţelor reale ale acesteia. (De exemplu programul de culegere Ventura Publisher conţine teste de imprimantă: forme diferite, caractere albe pe fond negru.) Deci cumpărarea unei imprimante cu laser trebuie făcută cu mare atenţie şi după o documentare şi testare amănunţită.

Imprimanta laser cu Postscript

Sînt cele mai bune imprimante pe piaţa mondială în momentul de faţă (preţuri între 6000-12000 DM).

Procedeul de tipărire a caracterelor şi a imaginilor este asemănător cu cel utilizat la imprimantele cu laser obişnuite şi de accea au aceleaşi avantaje:
- calitate de tipărire ridicată
- zgomot redus

Diferenţa constă în faptul că imprimanta este comandată de calculator printr-un limbaj special, de imprimantă, Postscript.

La aceste tipuri de imprimante caracterele se formează numai în perioada de pregătire pentru listare, ca urmare se realizează o economie de memorie atît în calculator cît şi în imprimantă. Mărimea caracterelor se poate alege opţional în timpul culegerii.

La imprimantele Postscript performanţele de imprimare sînt determinate de memoria operativă (RAM) pe care o au la dispoziţie şi de soft-ul rezident care execută comenzile Postscript. Pentru realizarea unei imagini grafice este nevoie de o memorie minimă de 1,5 MByte RAM, necesară atît la transferul caracterelor de la calculator cît şi la construcţia paginii.

Avantaje:
- utilizatorul poate alege din 35 de tipuri (standard) de caractere de diferite mărimi
- caracterele, desenele pot fi rotite
- se pot crea foarte uşor cercuri, pătrate şi alte forme geometrice

Dezavantaj:
- imprimanta trebuie să convertească comanda Postscript în sute de mii de mici puncte negre pe hîrtie, ceea ce duce la micşorarea vitezei de transfer.

Deoarece inventatorii, (firma "Adobe" din S.U.A.), cer o taxă de licenţă pentru fiecare imprimantă Postscript, preţul acestora rămîne ridicat în comparaţie cu imprimantele laser obişnuite. Iar varianta color a unei imprimante laser cu Postscript poate atinge 20.000 DM.

Imprimanta laser cu Postscript este un instrument de lucru pentru profesionişti:
- culegători pentru tipografie
- graficieni
- topografi
- specialişti în marketing şi în reclame.

7 sfaturi pentru o instalare corectă a imprimantei

1. Puneţi imprimanta pe un suport stabil. Mişcarea rapidă a capului de scriere, mai ales la imprimantele cu ace, poate răsturna mesele uşoare.

2. Imprimanta să fie instalată într-o încăpere apărată de soare. Folosirea imprimantei aflată sub razele soarelui, un timp îndelungat, poate duce la scăderea randamentului, cauzată de încălzirea excesivă a pieselor mecanice.

3. Imprimantele cu ace produc un zgomot mare la imprimare. Punînd sub imprimantă un suport din material sintetic spongios, de grosime de 2 cm., reduceţi simţitor zgomotul. Dacă se listează des şi în mod profesional se recomandă achiziţionarea unei carcase pentru protecţia împotriva zgomotului (preţ: 300 DM).

4. Înainte de a conecta imprimanta la calculator deconectaţi-o de la reţea. Tensiunea de alimentare cît şi electricitatea statică proprie pot distruge circuitele de interfaţă.

5. Dacă nu reuşiţi să listaţi trebuie verificat dacă conectoarele cablului de date atît la calculator cît şi la imprimantă sînt fixate bine cu şuruburi.

6. Hîrtia care urmează a fi utilizată la listare trebuie pusă pe masă şi nu pe jos. Dacă imprimanta trebuie să tragă hîrtia în sus un metru atunci există posibilitatea ruperii sau tractării necorespunzătoare a hîrtiei.

7. Deoarece imprimantele cu laser emană ozon (nociv) în timpul funcţionării, încăperile mici trebuie bine aerisite.

7 sfaturi pentru achiziţionarea de imprimante.

1. Să ştiţi exact la ce tipuri de listări veţi utiliza imprimanta. în primul rînd puteţi consulta vînzătorul, iar în al doilea rînd alţi utilizatori cu experienţă.

2. Trebuie să reflectaţi dacă doriţi să cumpăraţi de la un furnizor autorizat sau de la un "discounter" (angrosist), deoarece există diferenţe mari de preţuri. De aceea o discuţie înainte de cumpărare şi despre asistenţa tehnică este bine venită.

3. Consultînd preţurile de la mai mulţi furnizori autorizaţi sau discounteri puteţi realiza economii importante.

4. Nu întotdeauna modelele noi sînt mai bune, dar sigur că sînt mai scumpe. Deci, să nu vă fie teamă de modelele mai vechi. Astfel se poate cumpăra ieftin o imprimantă foarte bună.

5. Consultaţi mai multe oferte de preţ; astfel o să aveţi o privire de ansamblu mai bună asupra pieţii. Lăsaţi pe vînzător să vă demonstreze modul de funcţionare al imprimantei. Important: Nu vă grăbiţi ! Cumpăraţi în linişte !

6. Dacă aţi cumpărat o imprimantă, verificaţi încă în magazin că nu-i lipsesc accesoriile.

7.Nu uitaţi să cumpăraţi cablu de conectare la calculator. Acesta de obicei nu este inclus în preţul imprimantei.

ing. Zoltan Egyed

Cele mai cunoscute imprimante

Producător / Produs	Model	Tipuri caractere	Interfață p.	Interfață s.	Format hîrtie A4	Format hîrtie A3	Color cont	Compatibil EPS	Compatibil HP-L	Compatibil IBM	Preț DM
Brother											
HL8PS	L	11	da	da	da	nu	nu	nu	da	da	da	9700,-
HL8D	L	7, B	da	da	da	nu	nu	nu	da	da	da	1000,-
HL8e	L	7, 3xB	da	da	da	nu	nu	nu	da	da	da	6400,-
HL4	L	B(2x),L,G.BRO1/2	da	da	da	nu	nu	nu	da	da	da	3900,-
M3524L	24A	B,A,G,Q	da	da	da	da	da	nu	da	nu	da	4300,-
M2524L	24A	P	da	da	da	da	da	da	da	nu	da	3400,-
M1824L	24A	P,B,G,Q	da	da	da	nu	da	da	da	nu	da	1800,-
M1224L	24A	P	da	nu	da	nu	da	nu	da	nu	nu	1000,-
M2518	18A	P	da	nu	da	da	da	da	da	nu	da	2850,-
M1918	18A	P,Q	da	da	da	da	da	da	da	nu	da	1700,-
M1818	18A	P,Q	da	da	da	nu	da	da	da	nu	da	1500,-
M1209	9A	P,G,Q	da	da	da	nu	da	nu	da	nu	da	700,-
Citizen											
ProDot 9	9A	T,R,SS,	da	opț.	da	nu	da	opț.	da	nu	da	1500,-
ProDot 9x	9A	T,R,SS	da	opț.	da	da	da	opț.	da	nu	da	1900,-
120D+	9A	C,SS,R	da	opț.	da	nu	da	nu	da	nu	da	600,-
124D	24A	C,T,R	da	opț.	da	nu	da	nu	da	nu	da	800,-
Prolaser 12	L	C,L,R,T	da	da	da	nu	nu	nu	da	da	da	7000,-
C.Hoh											
C-240	9A	SS,R,S	da	da	da	nu	da	nu	da	nu	da	1700,-
CI-5000	18A	C,E	da	da	da	nu	da	opț.	da	nu	da	6850,-
C-610	24A	C10,12,14	da	da	da	nu	da	nu	da	nu	da	2250,-
C-645	24A	D,C,SS,R,G	da	da	da	da	da	nu	da	nu	da	3650,-
Epson											
LX-400	9A	R,D,SS	da	nu	da	da	da	nu	da	nu	nu	650,-
LX-850	9A	R,D,SS	da	nu	da	da	da	nu	da	nu	nu	800,-
FX-1050	9A	R,D,SS	da	nu	nu	da	da	nu	da	nu	nu	1850,-
LQ-400	24A	R,D	da	nu	da	nu	da	nu	da	nu	nu	900,-
LQ-860	24A	R,D,SS	da	da	da	nu	da	nu	da	nu	nu	2500,-
LQ-2550	24A	R,SS,P,C,OCR-A,B	da	da	nu	da	da	da	da	nu	nu	4500,-
SQ-850	24C	R,SS,D,P,COCR-A,B	da	da	da	nu	da	nu	da	nu	nu	2600,-
GQ-500	L	13 tipuri	da	da	da	nu	nu	nu	da	da	nu	3950,-
Kyocera											
F800	L	79 tipuri	da	da	da	nu	nu	nu	da	da	da	4700,-
F1000	L	"	da	da	da	nu	nu	nu	da	da	da	5950,-
F1800	L	"	da	da	da	nu	nu	nu	da	da	da	1050,-
F2000	L	35 tipuri	da	da	da	nu	nu	nu	nu	da	da	14800,-
Star											
LC-10	9A	SS,C,OR	da	opț.	da	nu	da	nu	da	nu	da	580,-
LC-10	9A	SS,S,OR	da	opț.	da	nu	da	da	da	nu	da	750,-
LC 24-10	24A	D,4LQ	da	opț.	da	nu	da	nu	da	nu	da	900,-
XB 24-10	24A	D,11LQ,	da	opț.	da	da	da	opț.	da	nu	da	2060,-
Laser Print 8DB	L	C,P,TR,LF	da	da	da	nu	nu	nu	da	da	da	7000,-
Laser Print 8DX	L	C,P,TR,LP	da	da	da	nu	nu	nu	da	da	da	8000,-
Siemens											
HighPrint											
PT 18	18A	D LQ	da	da	da	nu	da	nu	da	nu	da	2000,-
4400e	24A	E,Pi,LQ	da	da	da	da	da	da	da	nu	da	4200,-
PT 88S	C	6 tipuri	da	da	da	nu	da	nu	opț.	nu	opț.	2570,-
PT 10	L	6 tipuri	da	da	da	nu	nu	nu	nu	da	nu	6370,-
Oki											
ML 320 Elite	9A	C,SS	da	opț.	da	nu	da	nu	da	nu	da	1400,-
ML 32 Elite	9A	C,SS	da	opț.	da	da	da	nu	da	nu	da	1800,-
ML 380	24A	C,D,OR,H	da	opț.	da	nu	da	nu	da	nu	da	1109,-
OkiLaser	L	C,LP,H,TR	da	opț.	da	nu	nu	nu	nu	da	nu	3000,-

Legendă:
B = Brougham, G = Gothic, A = Anglia Q = Quatro, P = Prestige,
SS = Sans Serif, C = Courier, LP = Line Printer, D = Draft
TR = Times Roman, O = Orator, H = Helvette, E = Elite, P = Pica

Utilizarea raţională a memoriei calculatoarelor personale

În numărul 1/90 al revistei "if" am prezentat metode de extindere a capacităţii fizice a memoriei calculatoarelor personale, cît şi posibilităţile de adresare ale microprocesoarelor 80286 şi 80386 în "Real-mode" şi "Protected -mode". Revenim cu un articol în care vom prezenta mai în detaliu felul în care putem utiliza diferitele categorii de memorie, insistînd asupra memoriei extinse - EXTENDED MEMORY.

În general există trei categorii de memorie utilizate de calculatoarele personale compatibile IBM-PC:

- 1.Convenţională
- 2.Expandată (EXPANDED MEMORY)
- 3.Extinsă (EXTENDED MEMORY)

1. Noţiunea de memorie CONVENŢIONALĂ este legată de sistemul de operare MS-DOS şi se referă la memoria de lucru cuprinsă între 0 şi 640 K, fiind utilizată pentru rularea programelor DOS şi de aplicaţii. Restul de 384 KB memorie, pînă la 1024 KB (1MB) cît pot adresa microprocesoarele 8088,8086 şi 80286/80386 în "Real -mode", sînt rezervaţi pentru memoria de ecran a plăcilor grafice, sau memoria ROM (BIOS sau I/O). Pentru a putea trece peste limita celor 640 KB utilizabili sub MS-DOS există cele 2 metode prezentate în continuare.

2. Noţiunea de memorie EXPANDATĂ a rezultat din standardul introdus de firmele Lotus/Intel/ Microsoft (LIM) şi denumit EMS (EXPANDED MEMORY SPECIFICATION). Conform acestui standard se poate utiliza zona de memorie fizică cuprinsă între 640 KB - 1 MB, şi peste 1 MB pînă la 32 MB, (la 80286/80386 este aplicabil numai în "Real-mode"!) prin tehnica denumită "page switching"-comutare de pagină. Această tehnică divide segmente de 64 kbyte din memoria calculatorului în pagini de 16 kbyte, fiecărei pagini fiindu-i atribuită o adresă unică. O asemenea pagină poate fi "suprapusă" peste o zonă neutilizată din domeniul de 384 KB, apărînd astfel în domeniul memoriei adresabile de pînă la 1 MB. Existînd posibilitatea de comutare a acestor pagini, se poate utiliza practic întreg spaţiul de memorie existent (pînă la max. 32 MB!). (pag 78 PC Magazin Nr.47/90) Această metodă necesită atît un hardware adecvat (plăci speciale de memorie expandată care conţin o logică de comandă specială, pentru a putea suprapune paginile de memorie în spaţiul de adresare dorit şi care să permită schimbarea efectivă a adreselor de memorie) cît şi software realizat în acest scop (un driver de memorie, denumit Expanded Memory Manager - EMM). Versiunea actuală a acestor "administratori" de memorie expandată se numeşte LIM EMS 4.0. Acesta pune şi la dispoziţia programatorilor mai multe funcţiuni, prin intermediul întreruperii 67 hex.

Memoria expandată poate fi şi simulată, pe calculatoare care nu au memorie multă şi pe care se doreşte totuşi rularea unor programe mai mari care nu pot opera fără memorie EMS. Pentru aceasta există emulatoare EMS speciale care utilizează ca memorie expandată memoria extinsă, sau hard disk-ul.

3. Noţiunea de memorie EXTINSĂ a apărut odată cu utilizarea microprocesoarelor 80286/80386, care au extins posibilitatea de adresare fizică a memoriei operative la 16 Mbyte şi chiar 4 Gbyte. Acest lucru permite extinderea memoriei fizice prin instalarea de plăci de memorie suplimentare. Această extindere a memoriei a fost prevăzută pentru utilizarea unor sisteme de operare care să nu fie limitate de cei 640 Kbyte ai sistemului MS-DOS,cum sînt OS/2, XENIX şi UNIX, care pot administra 16 Mbyte de memorie.

Deoarece MS-DOS foloseşte modul de lucru al 80286, 80386, 80486 care se comportă ca şi

Funcţia 87 hex: Transfer de cuvinte între memoria de lucru şi memoria extinsă.
Intrare: AH = 87 hex
CX = Numărul cuvintelor de transferat
ES = Adresa de segment a GDT
SI = Adresa de offset a GDT
Ieşire: Bitul de carry (0-fără eroare; 1-cu eroare)
AX = Cod de eroare (0 - nici o eroare)

Funcţia 88 hex: Obţinerea capacităţii memoriei extinse
Intrare: AH = 88 hex
Ieşire: AX = Capacitatea memoriei extinse (în kbyte)

Funcţia 89 hex: Comutare în "mod protejat"
Intrare: AH = 89
Ieşire: nici una

Fig. 1 - Rutinele întreruperii 15hex, Protected Mode

8086, nu s-ar putea utiliza decît maxim 1 Mbyte al acestei memorii extinse.

Există totuşi posibilitatea de a utiliza această memorie estinsă şi sub MS-DOS. Spre deosebire de memoria expandată, accesul la memoria extinsă nu este normat. Totuşi s-au cristalizat două metode de utilizare a acestei memorii: metoda "întreruperii 15 hex" şi metoda Vdisk.

Cum se poate accesa memoria extinsă ?.

În principiu, utilizînd microprocesoarele 80286 şi 80386 în "mod real" - cum o face MS-DOS - acestea nu sînt altceva decît nişte 8086/8088 mai rapide, şi nu pot adresa decît 1 Mbyte de memorie. Noutatea adusă de 80286/80386 în această privinţă, o reprezintă posibilitatea funcţionării în "mod protejat". Acesta permite izolarea logică a domeniului de adresare pentru mai multe programe. Prin schimbarea rapidă între aceste programe este posibilă rularea în acelaşi timp (aparent) a mai multor programe. În "mod protejat" se pot utiliza 24 linii de adresă la 80286 şi 32 linii la 80386, putînd adresa astfel 16 Mbyte respectiv 4 Gbyte de memorie. Pentru a putea beneficia de aceste facilităţi oferite de microprocesoare, MS-DOS trebuie să comute funcţionarea în "mod protejat", pentru a accesa memoria extinsă dorită şi apoi să revină în "mod real". Acest lucru este posibil prin controlul unui bit de stare (flag) al CPU, dar utilizarea acestui procedeu este mai complicată decît pare la prima vedere. Cea mai simplă sarcină este de a comuta procesorul în "modul protejat". Pentru aceasta există instrucţiunea în cod maşina >lmsw<, dar mai trebuiesc avute în vedere şi anumite registre şi structurile de date, care în "mod real" nu au nici o utilizare, dar devin foarte importante în "mod protejat". Urmează apoi efectuarea accesului propriuzis la memoria extinsă, iar ultimul pas este şi cel mai greu: revenirea în "mod real", pentru a continua MS-DOS. Nu există nici o instrucţiune de revenire din "mod protejat" în "mod real" la 80286.

Abia la proiectarea lui 80386 s-a prevăzut acest lucru. Există totuşi o metodă de revenire în "mod real" şi la 80286: procesorul să primească un Reset, deoarece după un Reset procesorul se startează în "mod real". Dar după Reset calculatorul începe cu "self-test", ceea ce noi nu dorim în acest moment! Din fericire nu trebuie să vă îngrijiţi de chiar tot ceea ce ţine de aceste lucruri, deoarece BIOS-ul unui AT conţine cîteva funcţii pentru accesul la memoria extinsă. Aceste funcţii sînt disponibile prin intermediul întreruperii 15 hex, şi permit determinarea mărimii memoriei extinse, trecerea în "mod protejat", accesarea efectivă a memoriei extinse. (fig.1).

În principiu, prin intermediul întreruperii 15 hex se poate face tot ceea ce ţine de "protected mode". De un interes deosebit sînt primele două funcţii (87 hex si 88 hex) ale acestei întreruperi. Funcţia 89 hex nu este interesantă în acest moment, deorece utilizarea ei, fără a pregăti anterior cîteva registre importante, conduce la blocarea calculatorului.

Deplasament	Conţinut	Lungime
00 hex	rezervat	8 bytes
08 hex	Tabela de descriptori(GDT)	8 bytes
10 hex	Adresa sursă	8 bytes
18 hex	Adresa destinaţie	8 bytes
20 hex	Segment de cod BIOS	8 bytes
28 hex	Segment de stivă BIOS	8 bytes

Fig.2 - Structura GDT (Global Descriptor Table)

Funcţia 88 hex. Prin apelarea acestei funcţii se determină capacitatea memoriei extinse (în Kbyte) şi se returnează în registrul AX. În exemplele 1 şi 2 (xramchk.pas şi xramchk.c) se demonstrează cum puteţi folosi această funcţie în Turbo Pascal şi C. (listing 1 şi listing 2)

Funcţia 87 hex. Cu ajutorul acestei funcţii pot fi transferate blocuri de memorie din memoria convenţională în memoria extinsă şi invers. După cum se vede din fig.1, este necesară adresa de segment şi de offset a unei tabele denumită GDT (Global Descriptor Table). Microprocesoarele 80286 şi 80386 au fost proiectate să fie utilizate ca suport hardware pentru un sistem multitasking efectiv. Pe lîngă mecanismul protecţiei domeniului de adrese al unui program faţă de altul, ele prevăd şi procedeul de a memora, la nevoie, programe întregi sau segmente de program sau date pe harddisk, atunci cînd memoria operativă trebuie utilizată în alt scop. Cînd sînt necesare din nou aceste părţi de program sau date, ele sînt readuse în memoria operativă. Acest procedeu presupune existenţa unui mod de adresare a memoriei total diferit faţă de modul "real". Pentru a citi, de exemplu, în acest mod un byte din memorie se utilizează o adresă de segment şi una de offset, care împreună formează adresa fizică a celulei de memorie adresate.

În mod "protejat" un asemenea acces nu este posibil. Dacă un program ar avea aici posibilitatea să utilizeze o adresă fizică, ar putea să acceseze şi adrese din domeniul altui program. De aceea se utilizează numai adrese logice, iar transformarea adreselor logice în adrese fizice este realizată de către o unitate de administrare a memoriei (MMU = Memory Management Unit) integrată hard în procesor. Astfel conţinutul unui registru segment nu trebuie înţeles ca o adresă ci ca un "selector". Acesta este un indicator al unei tabele în care se găsesc la rîndul lor "descriptori".

Un descriptor descrie în final parametrii fizici ai segmentului de memorie utilizat, adică: -adresa fizică, lungimea segmentului şi drepturile de acces. Această tabelă, care conţine descriptori, se numeşte GDT.

În concluzie: -Un program care se rulează în mod protejat utilizează "selectori" pentru a accesa zona de memorie. -MMU determină, pe baza acestor selectori, descriptorii, numai după aceea procesorul putînd accesa adresa fizică. În cazul în care un domeniu de memorie a fost transferat pe un suport extern, descriptorul conţine un marcaj special, astfel încît MMU va încărca mai întîi acel domeniu de pe suportul extern în memoria internă. Cu ajutorul acestor selectori şi descriptori procesorul poate supraveghea toate accesele la diverse segmente de memorie.

Pentru a lucra în mod protejat trebuie mai întîi să pregătim deci o tabelă globală de descriptori (GDT), care să permită accesul atît la memoria convenţională cît şi la cea extinsă. Adresa acestei tabele trebuie definită în combinaţia de registre ES : SI Această tabelă GDT este valabilă pentru întreg sistemul de operare şi nu numai pentru un anumit program. (în acest caz ar fi vorba despre o tabelă locală de descriptori). (fig.2)

Importanţa cîmpurilor "adresă sursă" şi "adresă destinaţie." Aceste două cîmpuri conţin descriptori, care trebuie să indice adresele fizice ale segmentelor de memorie utilizate ca sursă respectiv destinaţie a transferului.

Fiecare descriptor se compune din 8 bytes. (fig.3)

Cîmpul "adresă fizică de segment" trebuie să conţină adresa fizică a segmentului de memorie de transferat (24 de biţi = 3 bytes)

Cîmpul "lungime segment" conţine lungimea zonei de memorie de transferat. Dar pentru BIOS trebuie să avem numărul de cuvinte de transferat (este bine să puneţi valoarea FFFF hex). Cîmpul "drepturi de acces" conţine informaţii despre: felul accesului permis (R, W, R/W), nivelul de privilegiu al programului, şi marcajul de "prezenţă în memoria internă". Este bine să prevedeţi valoarea 97 hex pentru acest cîmp.

Fig. 3 - Structura unui descriptor în cadrul GDT

Deplasament	Conţinut	Lungime
00 hex	Lungimea segmentului	2 bytes
02 hex	Adresa fizică a segmentului	3 bytes
05 hex	Drepturi de acces	1 byte
06 hex	Rezervat	2 bytes

Pentru a transfera cu ajutorul funcţiei 87 hex un bloc de memorie din memoria operativă în memoria extinsă, trebuie să pregătiţi doi descriptori in GDT. Apoi BIOS-ul copiază octeţii adresei de memorie indicate de descriptorul "adresă sursă" la adresa de memorie indicată de descriptorul "adresă destinaţie". Deci problema importantă la utilizarea funcţiei 87 hex constă în pregătirea unei structuri de date corecte în GDT. Am pregătit un exemplu (listing 3) "mutăram.asm" care arată cum pot fi transferaţi 1000 octeţi (500 cuvinte) de la adresa de memorie operativă 3000:0000 hex la adresa 0000:0000 hex din memoria extinsă.

La executarea acestei rutine, microprocesorul comută în "mod protejat" şi transferă numărul dat de cuvinte. Pentru a ajunge din nou în "real mode" se emite un semnal de Reset la procesorul 80286 de către funcţia 87 hex. Intrînd în self/test de iniţializare procesorul apelează un byte din memoria operativă, care a fost pregătit mai înainte acolo în acest scop. Acesta semnalizează procesorului că nu este vorba despre un "Cold-Start" ci de un Reset din "mod protejat". Drept urmare se efectuează o ramificare în partea de program a funcţiei 87 hex care urmează transferului de octeţi, astfel încît funcţia poate fi încheiată corect. Această metodă a fost denumită de către Gordon Letwin, conducătorul proiectanţilor de sistem ai firmei Microsoft, în cartea sa "Inside OS/2", drept <gararea automobilului la viteza de 100 Km/h>.

Un mic dezavantaj al acestei metode pentru utilizatorul de MS-DOS: La fiecare apel al funcţiilor întreruperii 15 hex, care comută în "mod protejat" se pierd un număr de semnale de întrerupere. În general acest lucru nu este prea rău, dar poate conduce la anumite probleme în programele care utilizează întreruperea de timer 1C hex, pentru a fi apelate periodic. Ceasul intern al MS-DOS este un asemenea program. La fiecare comutare în "mod protejat" şi înapoi se pierd cîteva ns. Dacă se utilizează des funcţiile din "mod protejat", timpul pierdut se cumulează la cîteva secunde. Dupa cum arată listing-ul 3, dacă se cunosc formalităţile la apelarea funcţiei 87 hex a întreruperii 15 hex, nu mai reprezintă o problemă schimbul de date între memoria operativă convenţională şi memoria extinsă. Înainte însă de a începe să transferaţi Kbytes de date să vă mai puneţi o problemă mare care apare odată cu accesul la memoria extinsă: BIOS-ul AT-ului şi al 386-PC-ului nu poate face o dinstincţie între diferitele programe care apelează memoria externă. Ce se întîmplă dacă mai multe programe utilizează în acelaşi timp memoria extinsă?

De exemplu un program ajutător rezident şi o aplicaţie de calcul tabelar, depun date în acelaşi domeniu de memorie extinsă deoarece "nu ştiu unul de celălalt". Începînd cu DOS versiunea 3.3 este şi mai rău: memoria extinsă poate fi utilizată ca RAM-disk. Cu DOS versiunea 4.0 se pot depune în memoria extinsă buffere de date

Deplasament	Conţinut în hexa
00 hex	59 22 29 00 00 00 00 00
08 hex	00 00 00 00 00 00 00 00
10 hex	00 00 00 00 00 00 00 18
18 hex	1c 21 3d 10 df 10 4f 01

Fig.4 Primii 32 de octeţi ai întreruperii 19 hex, fără RAM-disk instalat.

sau fişiere. Dar cine garantează că acestea nu utilizează exact aceaşi zonă pe care aţi utilizat-o în programul d-voastră ? Prima cerinţă este deci de a împiedica utilizarea de către un program a unei zone din memoria extinsă utilizată deja într-o aplicaţie. Pentru a ajunge la o soluţie, trebuie să vă puneţi întrebarea de unde ştie un program care zonă poate fi utilizată şi care nu. Există soluţii şi pentru aceasta! Mai întîi programul determină mărimea memoriei extinse prin intermediul funcţiei 88 hex a întreruperii 15 hex. Memoria semnalată de această funcţie este utilizată apoi total sau parţial.

Un program nu va utiliza deci o zonă de memorie care nu a fost confirmată de funcţia 88 hex, deoarece presupune că nu există altă zonă de memorie extinsă în afara celei semnalate. Deci pentru a realiza protejarea unei zone de memorie faţă de accesul altui program trebuie să vă asiguraţi ca BIOS-ul să nu returneze întreaga mărime a memoriei extinse, ci numai a domeniului utilizabil liber. Pentru aceasta trebuie să deviaţi vectorul disponibil pentru întreruperea 15 hex (adică adresa rutinei de deservire a întreruperii) într-o rutină proprie.

Să presupunem că aţi scris un program foarte avid de memorie, care necesită 640 Kb memorie extinsă. Dacă dispuneţi de 1 Mb de asemenea memorie, funcţia 88 hex vă returnează valoarea 1024 (Kb). Programul d-voastră trebuie să devieze întreruperea 15 hex într-o rutină proprie, care cuprinde funcţia 88 hex şi care semnalizează capacitatea memoriei extinse rămasă disponibilă, adică 1024-640=384 Kb.

Un program următor care apelează funcţia 88 hex va stabili o capacitate de 384 Kb a memoriei extinse şi nu va încerca să acceseze cei 640 Kb rezervaţi de d-voastră. La rîndul lui acest program poate deruta vectorul întreruperii 15 hex într-o rutină proprie şi să-şi rezerve în acelaşi mod o zonă de memorie necesară, lăsînd pentru programul următor şi mai puţină zonă de memorie extinsă.

Totuşi acest procedeu nu reprezintă o soluţie universală. O problemă apare la încheierea programului, cînd ar trebui eliberat spaţiul de memorie rezervat de acesta. Teoretic acest lucru se poate realiza prin redarea controlului vectorului întreruperii 15 hex către rutina iniţială din BIOS. Ce se întîmplă însă dacă între timp alte programe au utilizat acest vector ?. Rutinele de deservire a întreruperii acestor programe ar fi de asemenea evitate şi este relansată problema iniţială: un program nou încărcat va porni de la situaţia că dispune de întreaga memorie extinsă.

Mai există şi a doua problemă. Ce se întîmplă cu rutinele ajutătoare ale MS-DOS-ului, care utilizează memoria extinsă?. Un exemplu îl reprezintă driverul de RAM-disk <vdisk.sys> sau <ramdrive.sys>, care utilizează memoria extinsă. Pentru a utiliza deci memoria extinsă trebuie mai întîi să verificaţi că nu a fost instalat un RAM-disk. Dacă există, atunci trebuie să determinaţi capacitatea acestuia, iar spaţiul de memorie rămas puteţi să-l utilizaţi d-voastră.

Pentru a determina acest lucru trebuie să ştiţi

Cum funcţionează un RAM-DISK

Acesta se bazează pe derutarea vectorului întreruperii 19 hex într-o rutină proprie. În mod normal această întrerupere este utilizată la procedura de bootstrapare a sistemului. Noua rutină a driverului de RAM-disk nu face în principiu altceva decît să redea controlul la rutina de deservire a "vechii" întreruperi 19 hex.

Ca orice vector de întrerupere şi noul vector al întreruperii 19 hex se compune din: adresă de segment şi de offset.

Adresa de segment indică începutul segmentului de memorie pentru memoria extinsă de deservire a întreruperii iar adresa de offset este cea a primei instrucţiuni din această rutină. De cele mai multe ori adresa de offset este 0, deoarece începutul segmentului este în acelaşi timp şi începutul rutinei. În cazul prezentat, startul rutinei de deservire a întreruperii nu se găseşte totuşi la începutul segmentului, unde se află cîţiva octeţi de recunoaştere care arată dacă este instalat un RAM-disk. În continuare urmează prima şi în acelaşi timp ultima instrucţiune a noii rutine de deservire, care apelează din nou vechea rutină a întreruperii 19 hex. Efectul global al noii

Deplasament	Conţinut în hexa	Conţinut în ASCII
00 hex	00 00 e7 19 00 08 a9 00	- - - - - - - -
08 hex	d4 00 01 00 00 00 00 00	- - - - - - - -
10 hex	00 00 56 44 49 53 4b 20	- - VDISK
18 hex	20 56 33 2e 33 28 00 00	V3.3 - - -

Fig.5 Primii 32 de octeţi ai întreruperii 19 hex, cu RAM-disk instalat.

rutine, pentru întreruperea 19 hex, este acela că la începutul ei sînt depozitaţi cîţiva octeţi de recunoastere care indică instalarea unui RAM-disk. Fig.4 şi fig.5 arată ca exemplu primii 32 bytes şi întreruperea 19 hex.

Puteţi realiza această situaţie introducînd în fişierul <config.sys> următoarele: device = vdisk.sys 128/e.

După cum rezultă din fig.4 şi 5 prezenţa unui RAM-disk este indicată de conţinutul octeţilor 12 - 1C hex : VDISK V3.3.

Chiar dacă poziţia unde începe acest text diferă de la o versiune la alta a MS-DOS, dacă el se găseşte undeva în primii 32 bytes înseamnă că există un RAM-disk instalat. Nu este greu de imaginat o rutină care să determine prezenţa acestui text semnalizînd apoi prezenţa unui RAM-disk instalat.

Începutul acestui RAM-disk în memoria extinsă se poate găsi tot în primii 32 octeţi ai rutinei. Astfel la adresele 2C, 2D, 2E hex găsiţi valorile: 00, 00, 01. care reprezintă o adresă pe 24 de biţi, în format Intel: 010000 hex. Aceasta este adresa de start a RAM-disk-ului, deci imediat după limita de 1 Mb. În primii octeţi ai acestui RAM-disk găsiţi blocul de bootstrapare a RAM-diskului (în fig.6, primii 32 bytes). În octeţii 03 - 04 hex regăsiţi textul <VDISK V3.3> care indică existenţa unui RAM-disk instalat.

În octeţii 0B - 1D hex urmează blocul de parametrii BIOS, care conţine informaţii despre mărimea sectoarelor, a cluster-elor, membrul de FAT etc. În fig.6 avem instalat un RAM-disk care lucrează cu 128 bytes/sector, utilizează un sector/cluster, conţine un sector rezervat şi 1 FAT, permite maximum 64 de intrări în directorul principal şi are o capacitate globală de 1024 de sectoare = 128 Kb.

Octeţii 1E şi 1F conţin adresa primului octet, neocupat de RAM-disk, din memoria extinsă (reprezentată în Kbytes). În fig.6 recunoaşteţi valorile de 04 hex şi 80 hex, care dau adresa 0480 hex, ceea ce înseamnă că primul octet liber este la adresa 1024+128 Kbyte.

Cunoaşterea acestei adrese vă oferă şi posibilitatea să desemnaţi memoria extinsă ca ocupată, prin mărirea acestei valori.

Un program care urmează să folosească şi el memoria extinsă "vede" această zonă de memorie "rezervată" de programul d-voastră tot ca o zonă ocupată de RAM-disk. În felul acesta puteţi asigura protecţia datelor d-voastră din memoria extinsă faţă de alte programe.

Ce aveţi de făcut practic:

- căutaţi in primii 32 octeţi ai rutinei de serviciu a întreruperii 19 hex şirul de caractere "VDISK", pentru a stabili dacă este instalat sau nu un RAM-disk.
- dacă da, verificaţi dacă acest RAM-disk începe în memoria extinsă.
- dacă da, citiţi valorile de la adresele 1E hex şi 1F hex, adunaţi la ele zona de memorie pe care doriţi să o rezervaţi programului d-voastră şi scrieţi noua valoare în blocul de boot a RAM-disk-ului.
- în final, puteţi folosi zona de memorie rezervată astfel pentru rutinele sau datele programului d-voastră.

În concluzie, putem spune că există două procedee de a accesa memoria extinsă:

Primul, prin comutarea întreruperii 19 hex într-o rutină proprie, astfel încît programele care urmează să "vadă" că este mai puţină memorie liberă în memoria extinsă.

Al doilea, prin care se rezervă o zonă în memoria extinsă, modificînd blocul boot al RAM-disk-ului.

Observaţie: -deosebirea între cele două procedee este semnificativă: în primul caz se utilizează memoria extinsă de sus în jos, în al doilea caz de jos în sus.

Există şi o cale de mijloc (cea mai bună, de altfel!): mai întîi aflaţi prin intermediul întreruperii 15 hex mărimea memoriei extinse. Apoi verificaţi dacă este instalat un RAM-disk în memoria extinsă. În caz că da, aflaţi mărimea zonei de RAM-disk. Apoi trebuie să porniţi de la faptul că cea mai mică din valorile astfel aflate reprezintă valoarea adevărată a spaţiului de memorie disponibil în memoria extinsă, şi rămîne disponibil după RAM-disk.

Acest spaţiu de memorie puteţi să îl folosiţi pentru programele d-voastră . Mai trebuie doar să vă asiguraţi că alte programe nu vor avea acces la acest spaţiu de memorie. Pentru aceasta folosiţi o nouă rutină pentru întreruperea 19 hex care să modifice valoarea mărimii spaţiului de memorie extinsă cu valoarea dorită de d-voastră, atunci cînd este apelată funcţia 88 hex.

De asemenea trebuie să modificaţi şi valoarea mărimii "oficiale" a memoriei pentru RAM-disk, astfel încît zona ocupată de programele

Deplasament	Conţinut în hexa	Conţinut în ASCII
00 hex	00 00 00 56 44 49 58 4b	- - - VDISK
08 hex	33 2e 33 80 00 01 01 00	3.3 - - - - -
10 hex	01 40 00 00 02 fe 06 00	- - - - - - - -
18 hex	08 00 01 00 00 00 80 04	- - - - - - - -

Fig.6 Primii 32 de octeţi ai RAM-disk-ului.

d-voastră să pară ocupată de RAM-disk.

Dacă nu este instalat un RAM-disk, atunci depuneţi în memoria extinsă un număr de octeţi care să simuleze blocul boot al unui RAM-disk instalat. În acest fel programele d-voastră sînt păstrate în memoria extinsă cu toată siguranţa posibilă.

(ing. Emil Palade, ing. Iosif Fettich)

```
program  xramchk;
   (*  ------------------------------------------------------
       determina marimea memorie extinse
       (eXtended memory ) la un AT
       cu ajutorul functiei 88 hexa a INT 15 hexa.
       ------------------------------------------------------ *)
 uses dos;     (* biblioteca pentru accesul la UC *)
 var
     cpu: registers;
        (* variabile pentru acees la BIOS *)
    begin
        writeln; writeln;
        cpu.ah : = $88;  (* functia 88 hexa *)
        cpu.al : = 0;  (* AL sa fie nul  *)
        intr ($15, cpu); (* declanseaza intreruperea *)
        if (cpu.ax = 0)
        then begin
            writeln ('Calculatorul D-voastra fie este un XT,');
            writeln ('  fie nu are EXTENDED MEMORY instalata.');
            writeln;
        end
        else begin
            write ('Sistemul D-voastra poseda ',cpu.ax,
                   'kbyte Extended Memory.');
        end;
    end.
```

Listing 1 - rutina extram în Pascal

```
/*  ------------------------------------------------------
    XRAMCHK.C
    determina marimea memoriei extinse ( eXtended memory )
    la un AT prin intermediul functiei 88 hex a INT 15 hex
    ------------------------------------------------------ */
#include "dos.h"
integer extram()
/*  ------------------------------------
    functia intoarce marimea memoriei extinse
    ------------------------------------ */
{
    union REGS cpu;              /* variabile registre */
    cpu.h.ah = 0x88;             /* functia 88h        */
    cpu.h.al = 0;                /* AL sa fie nul      */
    int86 (0x15, &cpu, &cpu);    /* apel de BIOS       */
    return cpu.x.ax;             /* si retur           */
}
main()
{
    if (extram() = = 0)
    {
        printf ("\nCalculatorul D-voastra fie este un XT,\n");
        printf ("fie nu are EXTENDED MEMORY instalata.\n\n");
    }
    else
    {
        printf ("\n Calculatorul D-voastra poseda %d kbyte
                 Extended Memory.\n\n");
```

Listing 2 - rutina extram în C

```
; mutaram.asm
;
; muta un bloc de 100 de octeti de la adresa ;3000:0000  din  mem
; oria  conventionala  la adresa 0000:0000 din Extended Memory
;
;stiva    segment para stack
         dw  100h dup (?)
stiva    ends
;
date     segment para 'data'
gdt      db  16 dup(0)      ; rezervare pentru GDT
;
; adresa de origine:
         dw  0ffffh         ; lungimea segmentului
         db  0,0,3          ; adresa 30.000h
         db  97             ; drepturi de acces
         db  0,0            ; rezervat

; adresa destinatie
         dw  0ffffh         ; lungimea segmentului
         db  0,0,10h        ; adresa 100.000h
         db  97             ; drepturi de acces
         db  0,0            ; rezervat

; resetarea tabelei GDT
         db  16 dup (0)
date     ends

program  segment  para 'code'
         assume cs:program, ds:date, ss:stiva
start    proc  far ;
         mov  ax,date
         mov  ds,ax         ; incarca adresa segmentului de date in DS ;
         mov  cx,500        ; 1000 octeti = 500 cuvinte
         mov  ah,87h        ; functia 87 hexa
         push  ds           ; DS pe stiva
         pop  es            ; ES = DS

         mov  si,offset gdt ; adresa GDT
         int  15h           ; apel functie
         mov  ax,4c00h      ; termina programul prin MS-DOS
         int  21h
start    endp
program           ends
         end      start
```

Listing 3 - program de mutat o zonă de memorie

10 Base-T - Strategii de implementare

10 Base-T este noul standard pentru Ethernet pe cablu bifilar neecranat. Implementarea standardului ,cu multe funcţiuni noi, este foarte simplă, dacă se respectă anumite reguli.

În ultimele şase luni s-a ajuns la un aflux de prezentări de produse compatibile 10Base-T. Acestea au creat impresia că standardul 10Base-T este de mult omologat. De fapt aceasta s-a încheiat definitiv doar în septembrie. Utilizatorii care şi-au instalat deja o reţea Ethernet pe cablu neecranat torsadat (Unshielded Twisted Pair/UTP), s-ar putea lovi acum de probleme de transportabilitate şi de interoperativitate.

10Base-T este o specificaţie pentru Ethernet pe UTP cu o viteză de transfer de 10 MBit/sec. 10Base-T este o extensie a standardului Ethernet IEEE 802.3. Printre altele,specificaţia se referă la o topologie stea, o cablare strucurată, izolarea erorilor şi administrare centrală.. Instalarea ar trebui să fie în mod normal foarte simplă: cablurile UTP leagă nodurile reţelei cu distribuitoare centrale (Hubs) sau concentratoare aflate în dulapuri de cablare.. Fiecare nod al unui LAN 10Base-T dispune de un MAU (Media Access Unit).

Unii producători, ca de exemplu SynOptics, au oferit produse Ethernet UTP chiar înainte de elaborarea standardului. Ca rezultat <Lattis Net> nu este compatibil cu standardul. SynOptics nici n-a pretins acest lucru niciodată. Lattis Net foloseşte o altă schemă de integrare pentru legături ca şi un semnal de 2 V vîrf la vîrf (10Base-T prevede 5 V). În plus Lattis Net foloseşte aşa numita procedură de preegalizare în loc de postegalizare pentru a egaliza distorsiunile de semnal pe cablu. Lattis Net este totuşi un produs ce s-a dovedit a fi bun, cu o bază de instalări largă, şi SynOtics a dezvoltat o tehnică pentru integrarea 10Base-T în reţele Lattis Net. Alţi producători s-au angajat doar mai tîrziu pe piaţa Ethernet-UTP. Ei au avut avantajul că standardul exista deja într-o versiune preliminară. <Express Net> al firmei David Systems, de exemplu, a fost realizat compatibil cu 10Base-T.

10Base-T este compatibil cu celelalte standarde 802.3, ceea ce înseamnă că integrarea unei reţele LAN 10Base-T într-o reţea Ethernet existentă nu trebuie să fie neapărat complicată şi costisitoare. Există două metode de a lega o reţea 802.3 sau chiar nestandard UTP cu 10Base-T. O posibilitate este conectarea segmentului LAN printr-un aşa numit AUI (Attachment Unit Interface). Toate reţelele 802.3 sprijină această interfaţă. Astfel se pot conecta noduri cu diferite specificaţii Ethernet, ca de exemplu 10Base-5 şi 10Base-T. Interfaţa se va introduce între MAU şi Hub. Şi concentratoarele pot fi integrate în acest fel.

Produsele 10Base-T pot fi cuplate printr-un port AUI la un Hub sau MAU şi cu alte medii de transmisie (de ex. fibră optică). În plus se pot conecta Hub-uri prestandard prin AUI la Hub-urile standard.. Cu această metodă se obţine o reţea 10Base-T ierarhizată, care leagă mai multe reţele prestandard UTP-Ethernet în LAN. Distribuitoarele (Hub) 10Base-T pot fi utilizate la extinderea unui bus Ethernet existent, sau pentru a înlocui un repetor multiport. Astfel este posibilă, de exemplu, conectarea unui Hub 10Base-T la un segment coaxial sau din fibră de sticlă. În acest mod se pot utiliza cablurile telefonice existente pentru LAN-uri individuale.

În afară de metoda cu AUI mai există o a doua posibilitate pentru integrarea 10Base-T: Un modul de interfaţă se conectează direct la Hub sau concentrator. Aceste module posedă mai multe mufe pentru diferite metode de cablare, de exemplu coaxial sau fibră de sticlă. Asfel se pot cupla într-o topologie stea diferite segmente de cablu, ceea ce implică atît avantaje cît şi dezavantaje.

Această arhitectură permite integrarea tuturor mediilor de transmisie a segmentelor de reţea individuale pînă la Hub. Nu sînt deci necesare adaptoare pentru UTP. Comunicaţia dintre modulele gazdă va fi preluată de placa de bază (fundul de sertar) din Hub. Standardul 10Base-T nu furnizează specificaţii pentru Hub şi concentratoare, care dispun de un fund de sertar detaşabil. În principiu modulele prestandard Ethernet de interfaţă şi MAU pentru UTP sînt descrise în recomandări. Din această cauză produsele prestandard şi 10Base-T pot lucra împreună în acelaşi Hub. Avantajul acestei metode constă în faptul că protejează investiţiile deja făcute şi păstrează funcţionalitatea fiecărei părţi a reţelei. Dezavantajul este că volumul de muncă al administratorului de reţea, mai ales în LAN-urile mari, se măreşte, deoarece trebuie să supravegheze fiecare MAU şi să-i ataşeze modulul de interfaţă corespunzător.

În mod obişnuit legăturile prestandard şi cele 10Base-T nu funcţionează în cadrul aceluiaşi nodul Host (gazdă). Totuşi nu deranjează dacă se racordează un MAU prestandard la un modul 10Base-T. Din cauză că ambele componente utilizează tehnici de conectare diferite, nu recunosc faptul că există conexiunea şi nu transmit date. Trebuie să avem întotdeauna grijă ce tip de MAU se află conectat la ambele capete ale segmentului, înainte de a adăuga o legătură.

În prezent se lucrează la dezvoltarea unor proceduri care testează interoperabilitatea produselor 10Base-T. Aceste metode de test sînt dezvoltate în principal de asociaţiile de producători, cu toate că şi anumite instituţii independente lucrează la rezolvarea acestor probleme, de exemplu Universitatea din New Hampshire, Interoperability Laboratory (UNIOL).

Cu metoda UNIOL producătorii pot să-şi integreze produsele într-un mediu de la furnizori multipli, să determine şi să elimine problemele de interoperabilitate. Rezultatele testului sînt accesibile în orice caz doar membrilor unui consorţiu şi respectivilor producători. În testele UNIOL se testează în plus în principal interoperabilitatea şi nu respectarea standardului. Astfel este posibil ca produsele care trec aceste teste să fie integrabile într-un mediu 10Base-T şi cu toate acestea să nu fie conform standardului.

Pentru a elabora o strategie de integrare a 10Base-T într-o reţea Ethernet existentă trebuie să ţinem cont de cîţiva factori importanţi: printre altele mărimea şi felul bazei hard instalate, mărimea extensiilor planificate 10Base-T cît şi locul de amplasare.

Cea mai importantă este în mod sigur mărimea reţelei existente precum şi cea a extensiei dorite. Cînd partea instalată este relativ mică şi se are în vedere o extindere de amploare, cea mai avantajoasă soluţie poate fi înlocuirea echipamentelor existente şi reluarea de la început a implementării reţelei. Astfel se poate garanta o reţea omogenă conformă standardului. Dacă baza instalată este deja prea mare, schimbarea din prestandard în MAU-uri 10Base-T nu mai este necesară şi în plus este prea costisitoare.

Tipul Ethernet-ului prestandard UTP şi locul de amplasare al extensiei planificate sînt factorii următori ce trebuie avuţi în vedere. Cînd reţeaua urmează a fi extinsă într-o nouă clădire sau cu un dulap de conexiuni se recomandă metoda AUI sau legarea cu un modul gazdă (Host) la un Hub. În acest fel fiecare segment al reţelei este clar separat. Pe de altă parte dacă trebuie extins un segment deja existent se recomandă folosirea metodelor de integrare oferite de producătorul instalaţiei.

Cînd mediile 802.3 trebuiesc implementate mixt, mai există şi alte reguli care trebuie respectate, pentru a ne conforma specificaţiilor standardului. O parte a acestor reguli sînt conţinute deja în recomandările de pînă acum pentru 802.3, altele sînt stabilite în noile specificaţii 10Base-T. Secţiunea 13 a standardului "System Considerations for Multi-Segment Networks" (consideraţii de sistem asupra reţelelor multisegment), detaliază regulile care trebuie respectate cînd se implementează medii mixte. Aici aparţin: factori de întîrziere, numărul de repetoare între două terminale, lungimea maximă a segmentului de cablu.

Această parte a standardului 10Base-T furnizează informaţii despre realizarea unei reţele multisegment în cadrul unui aşa numit "Single Collision Domain" (domeniu de coliziune singulară). Ethernet se bazează pe metoda CSMA/CD (Carrier Sense Multiple Access/Collisin Detection). Domeniile de coliziune singulară se definesc ca reţele CSMA/CD independente, învecinate. Un repetor nu împarte reţeaua în domenii de coliziune singulară, această funcţie o realizează puntea (bridge). De vreme ce un Hub 10Base-T este un repetor, el nu împarte deci reţeaua în domenii independente ci doar îl extinde în principal.

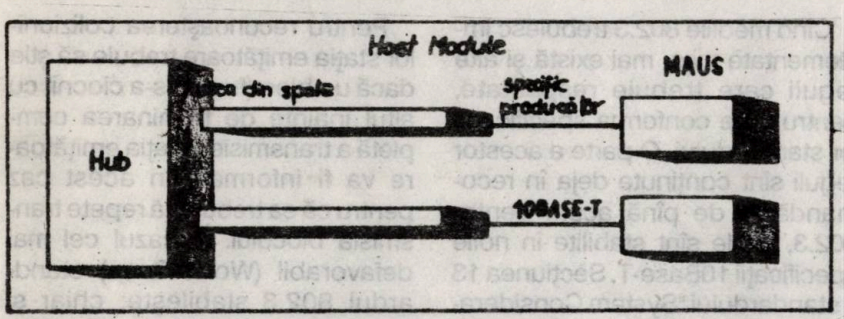
Recunoaşterea coliziunilor este o activitate importantă a metodei CSMA/CD şi întîrzierea este un factor hotărîtor, care trebuie avut în vedere la realizarea reţelei. Coliziunea totală, care se consideră într-un domeniu de coliziune, se stabileşte ca sumă a întîrzierilor ce rezultă de la MAU, repetoare, echipamente de introducere date şi mediul de transmisie. Fiecare din aceste componente produce un anumit factor de întîrziere, care totuşi este cunoscut în majoritatea cazurilor. Aceşti timpi trebuie calculaţi atunci cînd o reţea este redusă sau extinsă.

Pentru recunoaşterea coliziunilor staţia emiţătoare trebuie să ştie dacă un bloc (frame) s-a ciocnit cu altul înainte de terminarea completă a transmisiei. Staţia emiţătoare va fi informată în acest caz pentru că ea trebuie să repete transmisia blocului. În cazul cel mai defavorabil (Worst Case) standardul 802.3 stabileşte: chiar şi atunci cînd transmisia a parcurs întreaga reţea şi coliziunea a apărut în capătul cel mai îndepărtat, staţia emiţătoare trebuie să ia cunoştinţă de acest fapt, înainte de terminarea transmisiei.

În plus faţă de întîrzieri, numărul repetoarelor implementabile este influenţat şi de aşa numitele pauze între blocuri (interpacket gap) . Aceste pauze între blocuri sînt perioade de timp în care nu are loc nici un fel de transmisie. Cu ajutorul acestor pauze interbloc, aparatele pot identifica începutul şi sfîrşitul transmisiei. Lungimea maximă a segmentelor de cablu se află întotdeauna în legătură directă cu felul cablului utilizat. Fiecare clasă 802.3 are o lungime maximă recomandată de IEEE. De exemplu 10Base-2 este recomandată de Ethernet pentru banda de bază cu o viteză de trransmisie de 10 Mbit/sec şi o lungime de segment de 200 m.

Limitarea lungimii cablului este în legătură cu factorii stabiliţi, specifici pentru un anumit mediu de transmisie, de exemplu distorsiunea datelor la transmisie. Toate segmentele dintr-un mediu mixt trebuie să respecte recomandările prevăzute pentru respectivele medii. Aşa de exemplu un LAN

10Base 2 nu trebuie să aibă segmente mai lungi de 200 m. Mulţi producători dau pentru produsele lor o distanţă posibilă mai mare, totuşi trebuie avute în vedere lungimile care sînt prevăzute de specificaţiile IEEE. Doar astfel se poate realiza independenţa în perspectivă faţă de un anumit producător.



O instalare nouă Ethernet sau o extensie a unei baze instalate nu necesită o planificare minuţioasă. Cînd trebuie integrate componente 10Base-T într-o reţea Ethernet existentă sau într-un LAN mixt este important să realizăm în prealabil o diagramă a reţelei. Diagrama trebuie să cuprindă locul de amplasare al tuturor aparatelor precum şi lungimile şi felul cablurilor. La conectarea unor echipamente noi trebuie văzut, dacă au fost respectate recomandările Ethernet. Recomandările 10Base-T prevăd ca nici un echipament să nu se afle la o distanţă mai mare de o sută de metri de la Hub-ul corespunzător. După ce planificatorul reţelei s-a asigurat că această prescripţie este respectată, trebuie să calculeze factorii de întîrziere pentru fiecare componentă şi distribuirea repetoarelor. Calculele vor arăta de asemenea dacă are sens o punte sau un alt mediu de transmisie.

Definitivarea standardului 10Base-T a declanşat un aflux de anunţuri de noi produse. În plus a crescut în continuare acceptarea de Ethernet pe cabluri bifilare neecranate (UTP). Integrarea acestei tehnologii pare la prima vedere destul de complicată, dar este relativ simplă, dacă regulile de implementare sînt respectate.

(Greg Cambell)

Greg Cambell lucrează pentru Grupul Interconnect Consulting, o intreprindere care s-a specializat în planificarea, instalarea şi asistenţă pentru LAN şi WAN. Cambell conduce sucursala din San Francisco, California USA.

Unix şi viruşii

(continuare din pag. 15)

Împotriva exploatării slăbiciunilor sistemului, aşa cum a practicat-o cu succes virusul Internet, nu ajută decît grija permanentă asupra sistemului şi corectarea tuturor punctelor slabe devenite cunoscute. Prea multe puncte slabe le sînt cunoscute celor care administrează sistemul, dar nu sînt 'tratate' din comoditate sau din lipsă de simţ al responsabilităţii. Dacă o astfel de problemă nu poate fi corectată fără probleme, mai bine se renunţă la utilizarea softului respectiv.

Chiar la atacuri prin persoane care lucrează direct la sistem, utilizatorii autorizaţi sînt punctul cel mai nevralgic. În afară de deja pomenitele păcate, există o serie întreagă de obiceiuri care îi uşurează sarcina unui agresor. Astfel de exemplu deseori parolele sînt notate în locuri uşor accesibile, uneori chiar direct la terminal. Terminale nesupravegeate, unde sînt logaţi utilizatori privilegiaţi sau chiar administratorul sistemului, nu sînt nici ele neapărat o excepţie. Un intrus 'pe faţă' poate în cîteva clipe să îşi asigure şi pe viitor accesul. Abstracţie făcînd de paguba pe care un răuvoitor poate să o provoace uneori chiar şi cu o singură comandă. Aici şi administratorul Unix-ului poate să facă cîte ceva. Mediul multiproces al Unix-ului permite activarea unor aşa-zişi "demoni". Aceste programe pîndesc în background, devenind active numai cînd se întîmplă anumite evenimente. Astfel de exemplu un "idle"-demon, uşor de instalat, poate servi supravegherii utilizatorilor logaţi în sistem. Dacă mai mult timp nu se întîmplă nimic, sesiunea se termină în mod automat. Şi controlul activităţilor nocturne poate fi organizat în felul acesta.

Rezumînd, se constată că printr-o serie de caracteristici, cum ar fi uzuala conectare la reţea, Unix este mai expus atacurilor decît multe alte sisteme. Mecanismele de protecţie în forma de pînă acum sînt insuficiente şi au multe breşe. Arhitectura sistemului conţine însă posibilităţile necesare pentru a le aduce pe acestea la un nivel raţional pentru practică, fără un efort deosebit. Astfel treptele de securitate C1 şi C2 pot fi atinse în acest fel şi sînt deja realizate într-o serie întreagă de sisteme. Trepte de securitate mai înalte pot fi atinse în principiu, însă cu un efort considerabil. Faţă de MS-DOS sau chiar şi OS/2, Unix este sistemul mai sigur, chiar şi fără măsuri suplimentare. Cea mai slabă verigă în lanţul măsurilor de protecţie este însă utilizatorul - şi acest lucru e valabil nu numai pentru sistemele Unix.

Baze de date

Mult timp dBase III Plus s-a aflat în fruntea sistemelor de gestiune a bazelor de date (SGBD). dBase IV trebuia să întărească acest rol conducător. Prima încercare însă a dat greş, noua versiune lansată acum pe piaţă îşi propune să recucereasca tronul, dar concurenţa nu doarme.

Atunci cînd PC-urile au început să aibă succes, ele au fost folosite cu precădere în trei domenii de activitate: prelucrare de texte, calcul tabelar şi administrarea datelor. PC-urile însă nu erau destul de performante pentru cerinţele unui SGBD. Iniţial "bazele de date" nu erau decît simple cartoteci. Aplicaţiile care se bazează pe prelucrarea datelor - cum ar fi de exemplu facturarea sau contabilitatea - erau scrise cu ajutorul unor limbaje obişnuite de programare.

Aceasta s-a schimbat abia atunci cînd pe piaţă au apărut SGBD-uri puternice, care permiteau accesul programatorilor şi la colecţiile de date externe. dBase II era doar un simplu program de prelucrare a unei cartoteci, dBase II a fost totuşi un sistem puternic. Avantajul acestui program a fost limbajul său simplu. Sistemele anterioare erau scrise prea des de programatori, pentru programatori, utilizatorii fiind lăsaţi pe dinafară. dBase în schimb foloseşte un limbaj cu ajutorul căruia pînă şi neiniţiaţii se pot descurca intuitiv. Urmarea acestui avantaj a fost acapararea a mai mult de 60% din piaţă.

Odată cu creşterea performanţelor hardware-ului, s-au modificat şi cerinţele utilizatorilor. PC-urile îşi pierd rolul de "calculatoare personale" şi devin, datorită cuplării în reţele, tot mai mult, părţi ale sistemului informatic al unei întreprinderi. Pentru SGBD-uri aceasta înseamnă satisfacerea unor cerinţe sporite, mai ales în ceea ce priveşte posibilitatea de cuplare în reţea. Ashton-Tate s-a moşmondit prea mult cu această dezvoltare şi concurenţa a putut cîştiga teren.

Astăzi PC-urile au devenit o concurenţă serioasă pentru multe minicalculatoare. (Mainframe-urile nu sînt ameninţate, PC-urile neavînd încă puterea necesară. Pe de altă parte piaţa de mainframe-uri este saturată, aşa încît creşterea se va face în domeniul PC-urilor.) Sloganurile anilor viitori vor fi downsizing, cuplare în reţea şi concepte client-server.

Atunci cînd cei de la Ashton - Tate au observat că sînt pe cale "să rămînă de căruţă", şi-au propus să facă un salt cantitativ. dBase IV trebuia să fie nu doar urmaşul celui mai renumit SGBD, ci să marcheze şi "the-state-of-the-art" pentru mulţi ani. SQL-ul şi managementul tranzacţiilor sînt doar două din facilităţile preluate din lumea calculatoarelor mari. Compatibilitatea în jos, spre dBase III, trebuia însă să rămînă, pentru ca dBase să poată deveni veriga de legătură între aplicaţiile monoutilizator şi sistemul complex de administrare a datelor.

Prima încercare a dat greş, deoarece dezvoltarea lui dBase IV s-a făcut în pripă şi produsul a fost plin de erori. A urmat o dezvoltare de aproape doi ani, cu una din cele mai costisitoare faze de beta-testare din istoria prelucrării electronice a datelor. Acum a fost prezentat dBase IV 1.1. Dacă se ţine cont numai de caracteristicile de putere anunţate, acest program are toate atuurile să rămînă o referinţă pentru anii următori. Dacă însă nu-şi respectă promisiunile, atunci Ashton-Tate îşi poate scoate firma la vînzare. Mediul în care se mişcă dBase IV este multiplu. Alte SGBD-uri, cu concepte în parte contradictorii, încearcă să satisfacă diferitele cerinţe ale grupelor de utilizatori individuali. Utilizatorii care doresc să-şi administreze datele într-un mod mai simplu pot folosi sisteme conduse prin meniuri, ca de exemplu Dataease, cu care chiar şi neiniţiaţii, fără cunoştinţe de programare, pot ajunge la rezultate satisfăcătoare. Pentru administrarea unor volume mari de date există sisteme performante cum ar fi Paradox. Cerinţele programatorilor sînt satisfăcute de SGBD-uri cu compilatoare, cum ar fi Clipper şi Quicksilver, compilatorul SGBD-ului dBXL.

Versiunile PC ale marilor SGBD-uri, cum ar fi Ingres, Informix sau Progress, sînt concepute de producători doar ca versiuni de iniţiere. Ei îşi vînd sistemele cu suport mult şi nu găsesc în afacerea PC marjele necesare. În plus deservirea acestor sisteme, de multe ori, nu corespunde standardelor cu care ne-am obişnuit la PC-uri.

Drumul victorios al suprafeţelor utilizator grafice, început de Windows 3.0, încă nu a produs efecte notabile asupra bazelor de date. Pentru producătorii tradiţionali, ca Ashton Tate sau Borland, investiţia necesară pentru transpunerea sistemelor lor sub Windows este enormă. Dacă însă pentru SGBD-ul cu care se lucrează nu există o versiune Windows, atunci cheltuielile de transpunere sînt enorme pentru utilizatori, pentru că de regulă pot fi preluate doar fişierele de date, nu şi programele. De aceea SGBD-urile Superbase4 sau Omnis sînt folosite mai ales pentru proiecte noi.

Decizia de a utiliza un SGBD sau altul nu trebuie să depindă numai de performanţele acestuia, ci şi de problema care trebuie rezolvată. Nu programul cel mai bun este şi cea mai bună soluţie, ci programul cu ajutorul căruia problema pusă poate fi cel mai bine rezolvată.

(Computer Persönlich 20/90, Robert Bäurle)

dBase IV 1.1 - revenire reuşită

Chiar şi în cazul lui dBase, SGBD-ul care a impus la PC-uri standardul pentru această categorie de software, programul se adresează în principal la două categorii de utilizatori: utilizatorilor finali, care îşi pot administra foarte confortabil datele folosindul "centrul de control", şi proiectanţilor şi programatorilor care scriu aplicaţii cu dBase. După versiunea 1.0, care a avut mai multe defecte decît calităţi, Ashton Tate a lansat acum pe piaţă un produs lucrat cu migală şi testat profund. Ca orice alt software, nici dBase IV nu este fără cusur, dar este mai sigur şi mai performant.

Faţă de versiunea iniţială s-au făcut transformări în patru domenii: necesarul mai redus de memorie şi deservirea mai simplă se adresează în primul rînd utilizatorilor finali. Funcţiile şi comenzile create pentru obţinerea unei viteze de lucru mai mari, dezvoltarea limbajului de programare dBase, cît şi mai buna conlucrare dintre limbajul dBase şi SQL (Structured Query Language), se adresează în primul rînd programatorilor şi proiectanţilor. De aceste îmbunătăţiri ale performanţelor se bucură, bineînţeles, şi utilizatorii finali.

Prin folosirea mai economicoasă a memoriei, dBase IV necesită acum doar 450 KBytes, faţă de 516 KBytes cît necesita înainte. Se ajungea destul de des la probleme atunci cînd memoria trebuia împărţită între programe dBase foarte mari, drivere de reţea şi alte programe rezidente în memorie. Reducerea spaţiului de memorie utilizat are două aspecte pozitive: aplicaţiile individuale au la dispoziţie mai mult loc în memoria principlă şi simultan pot fi încărcate în memorie drivere de reţea mari.

Deservirea lui dBase a devenit mai confortabilă. Din opţiunile "Browse" şi "Edit", poate fi accesat direct meniul de administrare. Pe această cale se poate face o prelucrare rapidă de index. Pentru utilizator aceasta reprezintă o simplificare esenţială a lucrului, deoarece din cadrul acestui meniu el poate adăuga, modifica sau şterge foarte uşor indexuri. În versiunea anterioară acest lucru nu era posibil.

La conceperea unei interogări, în versiunea 1.0, lăţimea coloanei era limitată la 21 de caractere. Rezultatul unui QbE - interogare prin exemplu (Query by Example) - era prezentat pe ecran în maxim patru coloane. În versiunea 1.1 aceste limite au fost ridicate. Îmbunătăţirile din domeniile optimizării programelor şi tuning-ului se adresează în primul rînd proiectanţilor. Viteza de de lucru a programelor dBase poate fi sporită prin diferite metode.

Acum dBase conţine o opţiune de Disk-Cache, cu ajutorul căreia memoria extinsă sau expandată poate fi folosită ca buffer de memorie pentru accesul la harddisk. Dacă datele se află deja în memoria cache, atunci ele stau mai repede la dispoziţia prelucrărilor. Încă de la instalare se hotărăşte dacă se foloseşte memoria extinsă sau expandată ca memorie cache. Parametrii independenţi pot fi modificaţi şi ulterior. Dintre aceştia amintim parametrul "/w", care se foloseşte atunci cînd "se scrie direct pe harddisk", în loc să se folosească memoria cache. Dacă se doreşte modificarea parametrilor, atunci se poate lansa direct din dBase programul "dBCache". Programul Cache nu funcţionează decît împreună cu dBase IV. Trebuie să avem grijă, însă, ca în acelaşi timp să nu fie active şi alte programe Cache, deoarece aceasta ar duce la un dezastru al datelor.

Ashton Tate a conceput un sistem propriu de optimizare a administrării memoriei, denumit "sistem dinamic de-management al sistemului" - dMMS (dynamic Memory Management System) - în a cărui sarcină cade administrarea dinamică a overlay-urilor (a zonelor reacoperibile). Pe scurt: dBase-ul propriu-zis necesită 350 KBytes de memorie centrală, restul de 100 KBytes sînt rezervaţi administrării dinamice a zonelor reacoperibile. Zonele reacoperibile sînt părţi de dBase care nu trebuie să se afle în permanenţă în memoria centrală. Înafară de dBase în memoria centrală mai este necesar şi de loc pentru aplicaţii. În acest punct trebuie să se determine spaţiul necesar aplicaţiei, iar memoria centrală rămasă stă la dispoziţia administrării zonelor reacoperibile.

Pentru împărţirea spaţiului de memorie, dBase IV foloseşte variabila de mediu dBHeap. Cu această variabilă se poate stabili modul de împărţire procentuală a spaţiului de memorie între aplicaţii şi zonele reacoperibile. Variabila poate lua valori între 1 şi 100. Dacă se lucrează preponderent prin intermediul "centrului de control", atunci valoarea recomandată de Ashton Tate este dBHeap=10. În acest mod se rezervă mai mult spaţiu pentru zonele reacoperibile, ca de exemplu pentru măştile dBase şi pentru generatorul de rapoarte. Dacă dBHeap=90, atunci spaţiul este rezervat aproape integral aplicaţiilor. Deoarece dBHeap nu poate fi modificată din dBase, ea fiind o variabilă DOS, fixarea valorii ei trebuie făcută sub sistemul de operare, de exemplu în AUTOEXEC.BAT.

Ashton Tate a îmbunătăţit atît limbajul dBase cît şi cooperarea dintre acesta şi SQL. O comandă nouă este "Keyboard". Cu ajutorul ei pot fi scrise o serie de caractere în bufferul tastaturii. dBase poate prelucra apoi aceste caractere ca şi cum ar fi fost tastate de utilizator.

Ea poate fi folosită, de exemplu, la programe demonstrative care se parcurg singure, sau pentru selectarea automată a unui meniu. Pînă acum această comandă exista numai în dialectele dBase.

În mod asemănător ca şi la SGBD-ul "Foxpro", compatibil dBase, şi în dBase IV există acum posibilitatea "indexării condiţionate". Folosind o clauză FOR în comanda de indexare se poate formula o condiţie care să trebuiască a fi îndeplinită pentru ca un articol să fie preluat în index. Un exemplu de aplicare ar fi construirea unui index pentru abonaţii din Sălaj, şi nu pentru abonaţii din toată ţara. Avantaje: se face economie de spaţiu şi se măreşte viteza de căutare în index. Un index condiţionat nu poate fi folosit pentru fişierele cunoscute pînă acum în dBase, cu extensia .NDX, ci doar la cele cu extensia .MDX.

S-a extins şi domeniul de utilizare pentru funcţiile definite de utilizator (UDF) şi pentru comenzile ON. Pînă acum nu era posibilă executarea unor funcţii recursive cu funcţii definite de utilizator. Restricţiile existente pot fi ocolite incluzîndu-se în fişierul CONFIG.DB comanda "DBTrap=Off". Ce-i drept această operaţie trebuie făcută cu atenţie. Implicit dBTrap=On, şi valoarea variabilei ar trebui schimbată doar de programatorii cu experienţă care pot prevedea efectele acestei modificări.

S-au făcut modificări esenţiale şi în modul SQL al lui dBase IV. Cu noua versiune, în modul SQL pot fi lansate şi funcţii ale centrului de control. În modul SQL pot fi folosite acum şi comenzi dBase, nu doar comenzi SQL, iar în fişierele SQL pot fi inserate chiar şi funcţii scrise de utilizator.

În cazul instalării versiunii de reţea, dBase IV 1.1 poate fi folosit în următoarele tipuri de reţele: IBM PC Net începînd cu versiunea 1.2, Novell Netware 286, Novell Netware 386 şi 3Com3+ Share. În afara versiunii standard şi a celei de reţea mai există şi o versiune specială pentru programatorii şi casele de software care scriu aplicaţii cu baze de date.

Un programator îşi poate răspîndi aplicaţia sa împreună cu un modul Runtime, fără ca pentru aceasta să trebuiască să plătească licenţe la Ashton Tate. Producătorul afirmă că programele scrise cu această versiune sînt valabile şi pentru reţea. Pachetul de proiectare conţine în mod standard trei licenţe, aceasta permiţînd testarea confortabilă a aplicaţiilor multiuser. Această versiune costă 3819 mărci.

Demne de amintit sînt şi noile drivere de imprimantă, ca şi cele actualizate. Au fost adăugate drivere suplimentare pentru modelele noi de imprimante, printre ele numărîndu-se şi un driver Postscript pentru imprimanta Apple-Laser-Writer şi pentru imprimante compatibile Postscript.

Cu toate că dBase IV 1.1 stă acum la dispoziţia utilizatorilor, ei aşteaptă în continuare o ediţie server şi compilatorul dBase promis de Ashton Tate. După informaţiile obţinute în aceste domenii se lucrează zi şi noapte, dar cu noile cerinţe de calitate, autoimpuse, va trebui să mai avem răbdare.

(PC Magazin 43/90,
Jürgen Wasem-Gutensohn)

Paradox 3.5 - o punte între două lumi

Borland oferă cu Paradox 3.5 în combinaţie cu SQL Link unul din primele instrumente Front-End pentru servere de baze de date.

Sistemul de gestiune al bazelor de date (SGBD) Paradox, ocupă locul doi în lume, în domeniul bazelor de date pe PC-uri, după dBase. Acesta din urmă a pierdut în ultimii ani părţi importante din piaţă, atît în SUA cît şi în Europa. După părerea specialiştilor, în ultimul trimestru al anului 1990, Paradox a depăşit pentru prima dată dBase-ul ca număr de instalări noi.

Paradox este, în principal, un soft pentru tratarea confortabilă a datelor. Cu ajutorul lui pot fi făcute manipulări complexe şi cuprinzătoare de date chiar şi fără programare. Prima versiune Paradox a fost realizată în 1985. Sistemul are aşadar o tradiţie asemănător de lungă cu dBase-ul.

Un important semn de recunoaştere al Paradox-ului este tehnica Query-by-Example (QbE), un sistem interactiv de interogare. Într-un formular prezentat pe ecran sub forma unei tabele se marchează cîmpurile dorite, selecţia putîndu-se face făcînd uz de operatorii aritmetici şi logici obişnuiţi. La Paradox, ca şi la dBase, o tabelă este identică cu un fişier al bazei de date.

De facilităţi QbE, pentru analizarea şi manipularea datelor, dispune şi dBase-ul, dar forma de vizualizare tabelară a Paradoxului, este sub acest aspect mai convingătoare.

Paradox este singurul dintre bazele de date de marcă care conţine opţiunile standard de grafică integrate. Acelaşi lucru este valabil şi pentru prezentarea datelor sub formă de tabele încrucişate. Cu acestea informaţiile din tabele pot fi analizate asemănător ca la spreadsheet-uri.

Datorită tehnologiei VROOM (Virtual Runtime Object Oriented Memory Manager) şi componentului Turbo-Drive, după datele producătorului, Paradox 3.5 a devenit mai rapid şi mai eficient. VROOM, cunoscut deja de la programul de calcul tabelar Quattro Pro al firmei Borland, Paradox este descompus în componente mici, numite module. Aceste bucăţi de cod sînt încărcate în memorie numai atunci cînd este nevoie efectivă de ele. În acest mod în memorie rămîne mai mult loc pentru date, de exemplu pentru operaţiile de sortare mari consumatoare de memorie. VROOM preia în mod automat utilizarea memorie expandate sau extinse.

Turbo-Drive se îngrijeşte de adaptarea produsului Paradox la tipul procesorului disponibil, fie el un Intel 80286, un Intel 80386 sau un Intel 80486. Cu ajutorul acestei rutine, Paradox poate controla maxim 16 MBytes de memorie extinsă. Pînă acum acest lucru era posibil numai cu produsul Paradox 386, o versiune specială de Paradox optimizată pentru procesorul 386, care de la apariţia versiunii Paradox 3.5 a fost retrasă de pe piaţă. Datorită tehnologiei îmbunătăţite de management a memoriei, pusă la punct de Borland, Paradox poate lucra cu volume de date şi programe mai mari decît pînă acum. Dacă sistemul dispune de memorie extinsă, Turbo-Drive comută automat Paradox-ul în Protected Mode, cu urmarea că de acum înainte toate acţiunile Paradox se pot desfăşura direct în memoria principală. Performanţele se îmbunătăţesc în acest mod simţitor.

Setul de instrucţiuni şi funcţii oferit de PAL (Paradox Application Language, limbaj propriu pentru dezvoltarea de aplicaţii) a fost extins în noua versiune. Pentru dezvoltarea de aplicaţii mai complexe este oferit mediul Paradox-Engine. Dar cel mai important atu al Paradox-ului îl constituie şi pe mai departe tratarea confortabilă a datelor de către utilizatorul final. Produse asemănătoare în acest segment de piaţă sînt dBase, R:Base sau Dataease.

Prin intermediul limbajului de programare PAL şi Paradox dispune de unelte pentru dezvoltarea de aplicaţii. Comparativ cu limbajul dBase şi toate variantele lui, sau cu sistemul de dezvoltare al bazelor de date Clipper, PAL este utilizat într-o măsură mai redusă pentru elaborarea de programe. Pentru cine însă programarea este foarte importantă, este pus la dispoziţie mediul Paradox-Engine, un add-on, cu ajutorul căruia datele dintr-o bază de date Paradox pot fi accesate de aplicaţii scrise în limbaj C, cu Turbo-C sau Microsoft-C.

Paradox-Engine este gîndit pentru programatori şi conţine o interfaţă de dezvoltare a aplicaţiilor (API, Application Developement Interface), cu ajutorul căreia programatorii în C pot scrie aplicaţii care să poată utiliza tabelele Paradox în modul de operare stand-alone sau în reţea conlucrînd cu un file - server.

Făcînd o comparaţie între Paradox şi alte SGBD-uri cum sînt Clipper, dBase sau chiar R:Base, în ceea ce priveşte facilităţile pentru dezvoltarea de programe, acestea din urmă ies în avantaj. Pentru limbajele de programare ale acestora există compilatoare care protejează munca programatorului. Programele sursă odată compilate nu mai sînt vizibile pentru nimeni. Şi Paradox oferă o astfel de facilitate. Dacă un programator vrea să dea şi altora programele scrise de el în PAL, are nevoie de un modul Runtime. Licenţa pentru acest modul trebuie plătită însă separat şi costă 98 DM.

Enumerăm alte noutăţi aduse de Paradox 3.5: într-o machetă multitabelară pot fi cuplate nouă tabele, faţă de doar cinci tabele în versiunea anterioară, Paradox poate lucra acum şi în reţele 3+Open. În modul de operare stand-alone este suficientă o memorie de 512 KBytes, pentru lucrul cu Paradox. Împreună cu SQL Link, pentru accesul la servere IBM sau Microsoft este necesară o memorie de 1 MByte iar pentru accesul la un server Oracle este necesară o memorie de 1,5 MBytes.

Pentru cuplarea cu lumea exterioară este disponibilă o paletă bogată de formate de date, cum ar fi ASCII, dBase sau Lotus, în care datele pot fi exportate sau importate. Acelaşi lucru este valabil şi pentru programul de calcul tabelar Quattro Pro. Paradox poate citi fişierele scrise în formatul acestuia, sau poate scrie fişiere în formatul lui.

Cea mai importantă caracteristică a noii versiunii o constituie, pentru utilizatorul Paradox, programul auxiliar SQL Link. El lucrează numai împreună cu Paradox 3.5. Dacă într-o reţea sînt instalate ambele produse, comenzile SQL de interogare date, definire date şi manipulare date, sînt arătate automat în sistemul de meniuri al lui Paradox 3.5. Dacă SQL Link nu este instalat, punctele corespunzătoare din meniuri lipsesc.

Cu ajutorul acestui auxiliar, utilizatorii Paradox pot accesa într-o reţea datele SQL ale unei baze de date aflate pe server, de parcă ar fi vorba de tabele Paradox.

Produsul add-on oferă un acces transparent la date de la un IBM Database-Manager, Server Oracle, sau Server SQL Microsoft. După datele producătorului, vor urma extensii pentru a permite accesarea şi a altor tipuri de servere de baze de date.

Paradox SQL Link traduce comenzile şi interogările Paradox în instrucţiuni SQL, şi se îngrijeşte de comunicaţia fără probleme cu server-ul disponibil. Acordarea la serverul existent oferit de IBM, Microsoft sau Oracle, bineînţeles că este preluată automat de SQL Link.

Pentru a putea accesa, de la un post de lucru Paradox, datele aflate pe un server trebuie pornit mai întîi programul Set-up. El cercetează tabelele SQL ale serverului cu care este cuplat, şi face copii după toate aceste tabele pe staţia de lucru locală. Aceste tabele sînt goale dar au aceeaşi structură ca şi tabelele de pe server-ul bazei de date.

Manipulările de date care implică fişierele originale SQL, sînt efectuate mai întîi în copiile locale şi abia apoi sînt transmise prin SQL Link server-ului bazei. În acest mod sînt respectate şi toate cerinţele sistemelor tranzacţionale. Conform regulilor stricte ale acestora o succesiune logică de acţiuni nu se poate executa decît fie complet fie deloc. Respectarea principiilor sistemelor tranzacţionale în accesul în comun la colecţiile de date ale unui server de baze de date este necesară pentru asigurarea integrităţii datelor.

SQL Link traduce interogările formulate prin sistemul de meniuri Paradox, sau prin QbE, în instrucţiuni SQL. O funcţie plină de bun simţ este oferită de instrumentul ShowSQL, care afişează într-o fereastră forma SQL a cererii formulate prin QbE. Rezultatele interogărilor sînt prezentate pe ecran sub formă de tabele.

Din păcate, în versiunea actuală sînt neglijate interogările ce se adresează tabelelor SQL ale server -ului. Cereri care să permită combinarea de date aflate în tabelele Paradox locale şi în tabelele SQL Remote, sînt tot la fel de puţin permise ca şi formularea de cereri care să se refere la două servere de baze de date. Aceste restricţii vor fi ridicate abia în versiunile ulterioare. Cine doreşte totuşi să prelucreze împreună date aflate în tabele locale şi în tabele SQL, trebuie să formuleze întrebări distincte iar rezultatele să le stocheze local. Pe această cale ocolitoare, datele pot fi prelucrate local ca şi tabele Paradox şi din ele pot fi obţinute, de exemplu, şi grafice.

Paradox SQL Link lucrează cu comenzi interactive SQL şi cu instrucţiuni SQL imbricate în programe PAL. În programele PAL, ele sînt demarcate de cuvintele cheie SQL... ENDSQL. În acest mod pot fi scrise aplicaţii cu baze de date care se bazează pe arhitecturi client-server.

Paradox SQL Link este un instrument pentru analiza descentralizată a datelor şi evaluări de date care sînt memorate central pe un server de baze de date. Nici una din celelelalte SGBD-uri de pe PC-uri nu dispune în prezent de un astfel de instrument Front-End. Există astfel de produse anunţate de Ashton-Tate, Dataease şi Nantucket, dar pînă în prezent Borland este singurul producător care le şi oferă.

Datorită structurii sale Paradox nu are probleme cu SQL. Toate operaţiunile managementului SGBD-urilor se bazează pe o prezentare tabelară şi lucrează, ca şi SQL-ul, cu conceptul matematic al teoriei mulţimilor. Rezultatele definiţiei, manipulării şi interogării datelor sînt tratate ca o mulţime, în sens matematic. Spre deosebire de acest mod, dBase lucrează orientat pe articole.

Paradox Multipack înlocuieşte produsul Paradox Netpack, oferit pînă acum, şi este un pachet de programe pentru dezvoltarea unei baze de date în reţea cu un file-server. Preţul de 2793 DM a rămas neschimbat, Paradox 3.5 costă 2451 DM şi Paradox SQL Link costă 1600 DM.

(PC Magazin 47/90)

R:Base 3.1 - examen de maturitate

R:Base a apărut pe piaţă în 1984, primele versiuni numindu-se R:Base 4000 şi apoi R:Base 5000, pentru ca să se ajungă astăzi la R:Base 3.1. Iniţial produsul a fost dezvoltat de firma Wayne Ericson ca un sistem relaţional de tratare a informaţiilor, ca urmare a unei comenzi făcute de NASA în cadrul proiectului Shuttle. Pînă nu se sfîrşi încă anul, produsul era deja difuzat de firma Microrim. O dată cu noile versiuni Microrim a început din nou să desfacă produsul, de data aceasta însă sub regie proprie.

Versiunea standard a R:Base-ului pentru DOS, permite lucrul într-o reţea cu maxim trei utilizatori. Pentru mai mulţi utilizatori trebuie folosite programe auxiliare speciale. Produsul dispune de interfeţe care permit prelucrarea datelor R:Base din C, Pascal sau Fortran. Ca o extensie a generatorului de rapoarte integrat, se livrează programul Expanded Reportwriter, care permite o descriere foarte uşoară a raportului, evaluări matematice şi chiar şi funcţii pe şiruri de caractere. Pentru reprezentări grafice este livrat modulul DB-Grafix, asemănător programelor speciale de reprezentări grafice.

Operarea în R:Base se poate face în două moduri: prin intermediul meniurilor, sau în mod comandă. Pentru utilizatorii experimentaţi operarea în cel de-al doilea mod este mai comodă deoarece nu mai implică "plimbarea" prin meniuri pentru a se ajunge la funcţia dorită. Operarea orientată pe meniuri aduce aşa-numita tehnică de dare a comenzilor "Prompt by Example" (PBE) - dare a comenzilor după model - care împreună cu funcţiile

de Help şi cu manualul, permit utilizarea produsului şi de către utilizatorii neexperimentaţi. Este de asemenea permisă şi interogarea prin exemplu - QBE (Query by Example).

SGBD-ul dispune de generatoare care permit rapid şi simplu prin intermediul meniurilor crearea unei baze de date, construirea de aplicaţii, generarea de măşti şi elaborarea de rapoarte,. Produsul are implementat şi limbajul SQL, compatibil 100% ANSI nivelul 2 şi cu extensii DB2.

Sînt furnizate, de asemenea interfeţe care permit importul şi exportul de date dintr-o sumedenie de formate. Amintim doar formatele ASCII, dBase, Lotus, PFS şi PIF. Pentru asigurarea securităţii datelor pot fi folosite două comenzi: fie comanda SQL "Grant-Revoke", fie comanda "Read-Modify-File", cu ajutorul cărora tabelele şi formularele pot fi protejate în cazul accesului concurent la date.

R:Base dispune de două compilatoare. Primul poate traduce o aplicaţie în pseudocod şi este de fapt o soluţie Runtime, care permite economisirea de timp şi spaţiu, ca şi excluderea intervenţiilor nedorite în programe. Al doilea compilator permite convertirea programelor în fişiere executabile - .EXE -care pot fi rulate apoi independent sub sistemul de operare DOS. Aceste programe pot fi date mai departe fără să mai fie necesară plătirea unor licenţe. În cadrul aplicaţiilor compilate rapoartele se pot obţine de două, trei ori mai repede, măştile apar pe ecran cu viteza fulgerului, iar aplicaţiile pot fi rulate cu viteze de trei, cinci ori mai mari decît cu un interpretor. Programele sînt protejate, deoarece codul este transformat în limbaj maşină, şi întrucît utilizatorii finali nu mai trebuie să cumpere şi ei SGBD-ul putînd folosi direct aplicaţiile compilate elaborate de proiectanţi, ei vor economisi atît timp cît şi bani. Cu ajutorul compilatorului pot fi adăugate şi rutine scrise cu Microsoft C, sau cu Microsoft Assembler. Pentru depanare este livrat şi un DEBUGGER profesional.

Ce se mai "coace" la Microrim?

Microrim şi-a anunţat intrarea pe piaţă cu un nou SGBD Client / Server, care se va numi Vanguard şi a fost dezvoltat sub numele de cod Atlas.

Vanguard va fi prima "soluţie totală" şi va putea lucra sub OS/2 Presentation Manager, Windows, Apple Macintosh, DEC VAX/VMS, UNIX, MVS şi VM IBM. Versiunea pentru OS/2 trebuia să fie livrabilă în vara lui 1990, celelalte versiuni urmînd a fi terminate în următoarele 18 luni.

Acest SGBD va satisface cerinţele utilizatorilor în trei domenii: utilizare prietenoasă, interoperabilitate şi integritatea datelor. Utilizare prietenoasă: Vanguard va putea fi învăţat şi utilizat uşor şi rapid atît de începători cît şi de specialişti, atît de utilizatori cît şi de programatori. Interoperabilitate: Vanguard va putea fi rulat pe toate sistemele existente într-o întreprindere, şi va putea lucra şi împreună cu alte SGBD-uri existente deja. Integritatea datelor: Vanguard va proteja cea mai preţioasă comoară pe care o deţine o întreprindere - informaţiile sale - chiar şi în condiţiile în care la date au acces şi nespecialiştii.

Microrim a început dezvoltarea lui Vanguard, sub numele de cod Atlas, încă din mai 1987. David Hull, preşedintele firmei Microrim, este de părere că: "Vanguard este un nume ideal, pentru un software care vrea să arate direcţia în care vor evolua SGBD-urile generaţiilor viitoare. Cuvîntul Vanguard desemnează poziţia conducătoare într-un domeniu, într-o tendinţă. Vanguard va deveni prima soluţie pentru clienţii care caută o soulţie totală pentru schimbul de date într-o întreprindere." Vanguard s-ar putea traduce prin avangardă, pionier, înaintaş.

Esenţial la soluţia Vanguard este faptul că acest SGBD va putea lucra şi cu baze de date create cu SGBD-uri străine, cum ar fi de exemplu dBase, SGBD-ul IBM BS/2 Extended Edition, Oracle, DB2 şi altele. Aceasta înseamnă că Vanguard va putea citi, modifica şi scrie astfel de date. În acest fel bazele de date şi aplicaţiile existente într-o întreprindere vor putea fi utilizate în continuare, nu vor trebui modificate, şi vor putea fi legate între ele pentru efectuarea de schimburi de date..

(ing. Romulus Maier)

	dBase IV 1.1	Paradox 3.5	R:Base 3.1
Producător	Ashton Tate	Borland	Microrim
Preţ monoutilizator	2679	2451	2622
versiune reţea	3363	2793	2907
Necesar de memorie	640 KBytes	512 KBytes	640 KBytes
Meniuri pull-down	*	-	*
Număr articole	1 miliard	2 miliarde	nelimitat
Dimensiune articol	4000 Bytes	4000 Bytes	4096 Bytes
Număr cîmpuri	255	255	800
Precizia de calcul (zecimale)	20	15	15
Fişiere deschise simultan	99	nelimitat	80 tabele
SQL	*	cu SQL Link	*
Limbaj de programare propriu	*	*	*
Protecţie prin parolă	*	*	*

Compilatoare de baze de date

După cum se poate vedea din articolul precedent, SGBD-urile au devenit unelte confortabile de prelucrare a datelor. Cu ajutorul generatoarelor pot fi obţinute automat machete de introducere a datelor, etichete, rapoarte sau chiar aplicaţii întregi.

Aplicaţiile cu baze de date, care se bazează pe un SGBD, au avantajul că permit foarte uşor modificări şi că permit utilizatorului un acces direct la date, nu numai prin intermediul funcţiilor pe care i le pune la dispoziţie aplicaţia. Dezavantajul îl constituie prelucrarea relativ lentă a datelor. În afară de aceasta pentru fiecare aplicaţie în PC trebuie să existe SGBD-ul complet, sau cel puţin un modul Run-Time, ceea ce ridică costurile. Dar cel mai mare dezavantaj îl constituie faptul că atît datele cît şi programele sînt "deschise", oricine putînd avea acces la ele.

Compilatoarele de baze de date traduc limbaje de programare specializate, specifice unor SGBD-uri, în cod maşină. Datorită lor, la rulare, SGBD-ul propriu-zis nu mai este necesar, programul are o viteză de execuţie mai ridicată, iar datele şi sursa programului sînt protejate.

În continuare vă prezentăm trei compilatoare de baze de date: Clipper în noua versiune 5.0, Clarion Developer în versiunea actuală 2.1 şi Foxpro 1.0. Fiecare din cele trei programe a fost implementat de mii de ori şi şi-a dovedit calităţile atît în proiecte mici cît şi în proiecte mai mari. Dar cu toate că toate trei au fost dezvoltate pornindu-se de la aceleaşi cerinţe şi pot fi rulate pe acelaşi hardware, fiecare se bazează pe o filozofie proprie.

Foxpro şi Clipper au un trecut comun: ele sînt compilatoare dBase. Cu mai bine de trei ani în urmă încă alternative la marele înaintaş, în noile versiuni drumurile lor se despart. Foxpro urmează şi pe mai departe modelul dBase şi s-a dezvoltat într-un sistem atractiv care poate servi ca exemplu şi altor producători. Compilatorul integrat mai este încă o opţiune suplimentară a produsului program, dar nu mai este neapărat argumentul principal care să determine cumpărarea lui.

Cu totul altfel decît producătorii lui Foxpro, programatorii de la Clipper încearcă să se îndepărteze de modelul dBase şi să-şi dezvolte programul ca pe un limbaj de programare pentru aplicaţii cu baze de date de sine stătător, chiar dacă şi pe mai departe se bazează pe sintaxa dBase.

Clarion nu este înrudit cu dBase. Aceasta înseamnă că atît în ceea ce priveşte limbajul cît şi structura datelor urmează reguli proprii. Clarion a fost conceput de la început ca un sistem de dezvoltare de sine stătător şi urmăreşte cu consecvenţă să facă viaţa programatorului cît mai simplă cu putinţă.

Clipper 5.0

La obîrşie Clipper a fost o unealtă utilizabilă pentru compilarea programelor dBase pentru a se obţine din ele fişiere executabile (.EXE), care să poată fi rulate apoi şi în absenţa dBase-ului. Din acest motiv Clipper trebuise să preia cîteva "necurăţenii" ale dBase-ului. dBase este un interpretor care se concentrează asupra liniei curente din codul programului care se prelucrează. Dacă este apelat un subprogram, atunci dBase se "uită" mai întîi pe harddisk să vadă dacă acolo există un program cu acest nume. dBase stochează variabilele într-un spaţiu de memorie rezervat. Toate acestea contrazic, natural, modul de lucru al unui compilator: rezervarea pauşală de memorie pentru variabile este risipă curată de spaţiu. Din acest motiv în cazul compilatoarelor variabilele trebuie declarate exact, pentru a se putea rezerva numai spaţiul necesar. Un compilator nu se poate uita pe harddisk abia în momentul rulării pentru a vedea dacă o subrutină există sau nu. El necesită prezenţa tuturor părţilor programului în momentul compilării şi construieşte cu ajutorul lor programul final. În sensul acestei tratări diferite a subprogramelor, Clipper a permis încă de la început construirea mai multor proceduri în interiorul unui fişier .PRG. În afară de aceasta Clipper a cunoscut încă de la început aşanumitele funcţii definite de utilizator (User Defined Functions - UDF). Ambele facilităţi le-a preluat însă şi dBase în versiunea IV.

În versiunea 5.0, însă, Clipper merge pe drumul său propriu. Clipper 5.0 se aseamănă foarte mult cu limbajele de programare clasice. Tratarea variabilelor este un capitol important în acest sens. Risipa tocmai amintită de spaţiu de memorie, datorită principiului de repartizare a memoriei rezervate, nu a putut fi înlăturată în întregime pentru că s-ar fi pierdut compatibilitatea cu versiunile mai vechi de Clipper. Totuşi în Clipper 5.0, cu ajutorul unor directive, pot fi declarate variabile care să nu-şi schimbe valoarea pe lungimea întregului program. Compilatorul implantează aceste variabile direct în cod şi pentru ele nu trebuie rezervat un loc special în memorie. Acelaşi lucru se întîmplă şi pentru tipul declarativ "Static": şi în acest caz este vorba despre o variabilă care nu-şi schimbă valoarea în timpul rulării programului.

Şi fără aceasta însă în noua versiune a lui Clipper există diferenţe demne de consemnat în modul de tratare a memoriei. Astfel este utilizată o altă formă de transmitere a parametrilor unei proceduri. În sintaxa veche a instrucţiunilor cu

DO şi WITH, variabilele erau transmise prin referinţă, aceasta însemnînd că i se transmiteau procedurii adresele din memorie la care se aflau aceste variabile, procedura accesînd apoi direct respectivele zone de memorie. În cazul unei erori în program procedurile şi funcţiile puteau schimba aceste valori involuntar, deoarece erau accesate variabilele originale. În noua sintaxă, şi în cazul funcţiilor, variabilele sînt transmise prin valoare, aceasta însemnînd că procedurile şi funcţiile primesc o copie a variabilelor cu valoarea lor actuală, şi că nu mai au acces la original. Dacă este necesară, totuşi, transmiterea prin referinţă, atunci este suficient să se pună în faţa variabilei caracterul "@".

Un nou tip declarativ permite o mai bună supraveghere a variabilelor. Dacă o variabilă este declarată "Private", atunci din acest moment ea este valabilă pe nivelul curent şi pe toate subnivelele apelate din acest punct. O declaraţie Private în programul principal este sinonimă cu o declaraţie Public, deoarece toate celelalte părţi ale programului sînt apelate din acest punct. Prin noul tip "Local", din contră, o variabilă este declarată cunoscută numai în partea de program actuală, ea rămînînd necunoscută în subprogramele apelate. Pe lîngă controlul îmbunătăţit, aceasta mai are şi avantajul că permite utilizarea aceloraşi nume de variabile în fiecare parte a programului, atît timp cît acestea sînt declarate Local.

Noutatea esenţială o constituie însă driverele de baze de date interschimbabile. Clipper recunoaşte formatele fişierelor create cu dBase III, Paradox sau un Server SQL şi lucrează intern cu sintaxa dBase/Clipper.

Foxpro 1.0

Foxpro şi-a propus un cu totul alt scop. Proiectanţii şi-au pus problema: cum ar trebui să arate urmaşul lui dBase III Plus, şi au dotat programul cu tot ceea ce are nevoie un SGBD modern cu mediu de dezvoltare integrat. Rezultatul obţinut este un SGBD care are una dintre cele mai atractive suprafeţe utilizator, orientată caracter, din momentul de faţă, cuplat cu un compilator foarte puternic.

Foxpro a preluat structura de date a Foxbase-ului, care era compatibilă cu dBase şi care utiliza exclusiv fişiere indexate proprii. Noile tipuri de cîmpuri numerice "Floating Point" şi "Simplă precizie", apărute în dBase IV, au fost preluate. Noile fişiere de index .MDX, apărute în dBase IV, care permit pînă la 48 de indexuri pentru fiecare bază de date, rămîn caracteristice doar dBase-ului. Foxpro permite pînă la 21 de indexuri pentru un fişier, dar în total nu pot fi deschise mai mult de 25 de indexuri simultan. Şi Clipper renunţă la acest tip practic de fişier index, dar în noua versiune permite cîte 15 fişiere index pentru un domeniu de lucru, la o medie de 250 de domenii de lucru. Foxpro a păstrat chiar şi limitarea dBase de maxim 100 de caractere pentru o expresie de index.

Ca şi Clipper, nici Foxpro nu permite comenzi SQL adresate direct, aşa cum este permis în dBase IV. Sintaxa se bazează pe Foxbase şi cu aceasta pe dBase III, dar a fost extinsă cu multe comenzi folositoare. Sînt acceptate şi noile comenzi apărute în dBase IV, dar există şi comenzi noi.

	Clipper 5.0	Foxpro 1.0	Clarion 2.1
Producător	Nantucket	Fox Software	STF Software
Preţ (DM)	2730	2500	2680
Runtime	1930		
reţea	3415		
Necesar memorie	512 KB	512 KB	512 KB
Fişiere EXE	da	nu	da
Debugger	da	da	da
Ferestre	da	da	da
Operare cu mouse	nu	da	nu
Prelucrare bază de date	nu	da	da
Număr articole/fişier	1 miliard	1 miliard	nelimitat
Număr cîmpuri/articol	1024	255	nelimitat
Tipuri de cîmpuri:			
caracter	da	da	da
numeric	da	real	zecimal
			short, long, real
dată	da	da	nu
timp	nu	nu	nu
valoare	nu	nu	nu
logic	da	da	da
selecţie	nu	nu	da
memo	da	da	da
Număr fişiere deschise simultan	250	10	252
Indexuri / fişier	15	10	252
Compatibilitate			
comenzi dBase	dBase III	da	nu
fişiere dBase	da	da	anunţată
versiune Unix	nu	da	anunţată
versiune Macintosh	nu	da	anunţată

Suprafaţa atractivă este un mediu de programare confortabil. Editorul este integrat, se poate opera cu ajutorul mouse-ului şi pot fi prelucrate mai multe fişiere simultan. Cu ajutorul funcţiilor Cut şi Paste pot fi copiate informaţii din toate ferestrele simultan deschise.

Programul dispune şi de un depanator (Debugger) şi de un program de trasare (Trace Programm), integrate în mediul de dezvoltare, şi de programe ajutătoare cum ar fi generatorul de machete şi de programe, şi chiar de un program ajutător pentru documentarea automată a programelor.

Şi în cazul compilatorului Foxpro îşi urmează drumul propriu. În loc să elaboreze cod executabil complet (.EXE), se obţine un format intermediar care la rulare necesită existenţa lui Foxpro sau a unui modul Run-Time. Acesta nu este neapărat un dezavantaj, deoarece Foxpro este mai rapid cu 20% decît vechiul Clipper. Proiectantul însă trebuie să-şi bage mîna adînc în buzunar pentru a plăti şi modulul Run-Time. Această investiţie se face însă o singură dată, modulul putînd fi dat mai departe fără restricţii.

Spre deosebire de Clipper, Foxpro nu se limitează doar la lumea DOS, existînd versiuni şi pentru Macintosh şi Unix.

Clarion 2.1

Clarion îşi primeşte utilizatorii cu braţele deschise. În meniul introductiv sînt afişate toate instrumentele: instrumente de programare, mediu de programare şi partea de bază de date. Instrumentelor de programare le aparţin editorul, generatorul de programe, precum şi un program pentru generarea unor machete cu mesaje de ajutor.

În mediul de programare, programele elaborate pot fi rulate atît timp cît n-au fost transformate încă în fişiere .EXE cu un linkeditor extern. Clarion cunoaşte două medii: cu şi fără debugger (depanator). Depanatorul este folosit numai în timpul dezvoltării, programele definitivate sînt rulate apoi fără acesta.

În partea de bază de date sînt create şi modificate structurile fişierelor, sînt puse la dispoziţie comenzi pentru prelucrarea directă a datelor, dar mai ales aici se găsesc funcţiile de import şi export. Pe lîngă formatul propriu, Clarion mai poate citi şi scrie şi fişiere dBase, DIF şi Basic. În acest fel este posibil un schimb de date cu alte programe fără probleme.

Componenta Designer este un amestec între un program de asistenţă în proiectare şi un generator de programe. Nici Clipper şi nici Foxpro nu posedă ceva asemănător.

În principal există trei tipuri de subprograme pentru aplicaţiile cu baze de date: meniuri, machete şi rapoarte. Clarion a standardizat aceste tipuri şi le-a înlocuit cu alte două proceduri standard: tabele şi altele. Tabelele sînt tratate ca o unitate şi nu ca înregistrări individuale. Sub titulatura de "altele" se ascunde o interfaţă spre toate subprogramele care nu au fost elaborate cu Designer. Acestea pot fi rutine în limbajul Clarion scrise de utilizator, dar şi module scrise în alte limbaje de programare, sau module create cu Designer dar în care s-a intervenit cu editorul şi din acest motiv nu mai pot fi prelucrate în continuare cu Designer.

O caracteristică tipică Clarion este predefinirea tastaturii. La cîmpurile numerice, la dorinţă, este activă automat tasta Num-Lock, la cîmpurile de text poate fi prestabilit modul inserare sau suprascriere. Această caracteristică foarte folositoare conduce însă la o folosire diferită a tastaturii în cadrul fiecărui cîmp. Cine parcurge cu cursorul cîmpurile şi ajunge într-un cîmp numeric, se vede dintr-o dată confruntat cu un şir de 2-uri.

Sistemul Clarion a fost dezvoltat ca limbaj şi nu ca program de ajutor în prelucrarea datelor. Rezultatul este un limbaj, care conţine şi elemente de C (de ex. fişiere Include), foarte curat construit şi uşor de învăţat.

Facilitatea de lucru în reţea este elegant rezolvată. La generarea codului cu ajutorul instrucţiunilor lui Designer este dat un model de prezentare. Dacă în locul modelului standard se dă modelul reţelei, atunci codul obţinut poate fi rulat în reţea.

Cele mai noi concepte, cum ar fi suprafaţă utilizator grafică, arhitectură client-server, sau programarea orientată pe obiecte, nu sînt implementate. Sînt anunţate versiuni pentru OS/2, Unix şi Macintosh, dar încă nu sînt disponibile.

(Computer Persönlich 26/90, Robert Bäuerle)

Aplicaţii distribuite

(continuare din pag. 43)

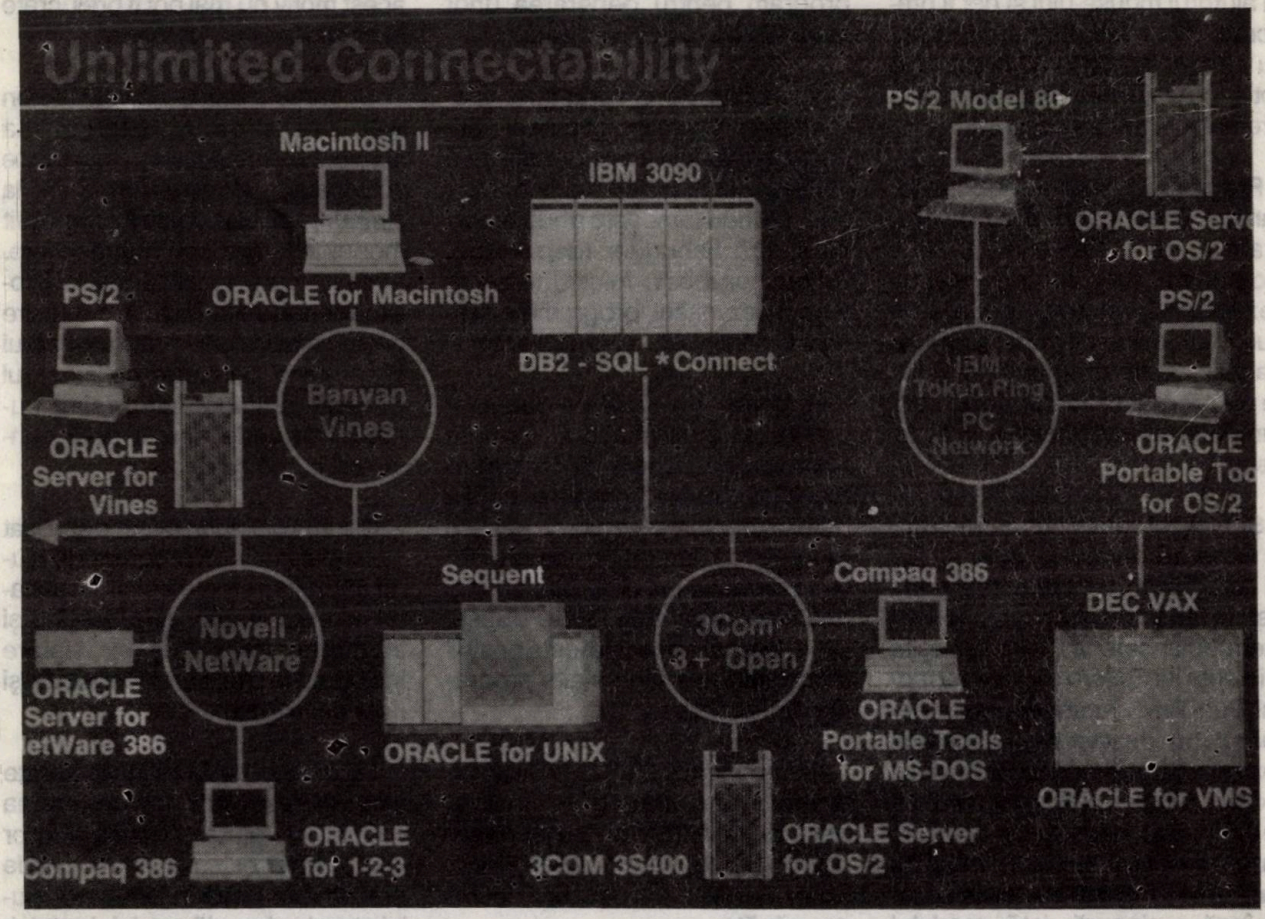
În practică trebuie să se mai ţină cont şi de multe alte aspecte suplimentare:

- faze de test detaliate
- instalare
- organizarea fluxului de lucru
- întreţinere şi - suport.

Prin intermediul aplicaţiilor distribuite programatorii pătrund în regiuni necunoscute pînă acum. Premisa esenţială este înţelegerea şi acomodarea cu noile instrumente şi tehnici de lucru.

(Computer Persönlich 22/90, Steve Guengerich, Dwight Williams)

Aplicaţii distribuite



Aplicaţiile distribuite utilizează reţelele în mod optim

Fundamentele prelucrării distribuite a datelor se pun în momentul conceperii design-ului. Vă oferim o privire de ansamblu asupra stării actuale a tehnicilor specifice şi vă prezentăm cîteva instrumente şi utilitare destinate dezvoltării de aplicaţii cu baze de date distribuite.

Tendinţa de a dezvolta aplicaţii distribuite este determinată în principal de dorinţa de a se exploata puterea de calcul a noilor PC-uri aflate pe birourile întreprinderilor private şi ale administraţiei publice. Aplicaţiile distribuite oferă în aceste cazuri avantaje importante: datorită păstrării distribuite a datelor în sistemul informatic al întreprinderii multe operaţii pot fi executate în paralel. Scriind aplicaţiile distribuite în aşa fel încît datele să se afle fizic acolo unde este cel mai des nevoie de ele, scad atît cheltuielile pentru comunicaţii cît şi costurile datorate lor într-o întreprindere.

Proiectarea şi dezvoltarea aplicaţiilor distribuite sînt probleme foarte complexe. Dar în aceeaşi măsură în care cresc cheltuielile, creşte şi numărul producătorilor care pun la dispoziţie instrumente pentru dezvoltare. Cei mai cunoscuţi producători de sisteme de baze de date distribuite sînt: "Empress", "Informix", "Ingres", "Oracle", "Sybase". Şi IBM oferă prin specificaţia APPC (Advanced Program to Program Communications) standarde pentru protocoale ale unei prelucrări distribuite şi rutinele necesare pentru a putea exploata protocoalele respective. Acestea, pentru a simplifica dezvoltarea aplicaţiilor distribuite, sînt integrate în sistemul de operare OS/2 Extended Edition, al lui IBM.

Din punct de vedere al modelului există trei forme de prelucrare a datelor: centralizat, descentralizat şi distribuit. Sistemele pur centralizate sînt folosite pentru aplicaţiile întreprinderilor mari, care se prelucrează pe mainframe-uri (calculatoare mari). Datele şi programele, în majoritatea cazurilor se află în acelaşi calculator. Sistemele descentralizate se pretează pentru aplicaţii care se rulează pe calculatoare distincte, fără să existe un schimb de date între o aplicaţie şi alta. Pe fiecare calculator se găsesc aceleaşi programe şi aceleaşi structuri de date, nu însă şi

aceleaşi date. Sistemele cu un singur utilizator (single-user) au apărut odată cu PC-urile. Cu toate că tendinţa de a utiliza reţele este tot mai mare, în continuare cele mai multe programe se scriu pentru sisteme single-user, mai ales pentru micii meseriaşi şi pentru întreprinderile mici din sfera serviciilor.

Sistemele distribuite sînt un amestec de aplicaţii centralizate şi descentralizate. Ele înglobează avantajele celor două forme anterioare. În cazul lor programele şi datele se pot găsi în orice loc: centralizat pe un mainframe, pe calculatorul dintr-o secţie, pe fileserver-ul unei grupe de lucru, sau pe PC-ul unui specialist.

Sistemele distribuite sînt cele mai apropiate de atingerea scopului - realizarea unui echilibru între o viteză mare de prelucrare a datelor şi realizarea unui control permanent asupra consistenţei datelor. Instanţele de control sub formă de backup, restart, recovery şi management al reţelei sînt centralizate în cazul lor. Viteza mare de lucru se atinge, în acest caz, datorită faptului că programele şi datele se află acolo unde sînt necesare cel mai des. Aceasta garantează timpi de răspuns scurţi la aplicaţiile zilnice.

Dezavantajul sistemelor de baze de date distribuite îl constituie costul mare al echipamentelor hard, deoarece aplicaţiile distribuite au nevoie de inteligenţă proprie şi de o memorie externă relativ mare. Şi costurile pentru soft pot fi mari, dacă aplicaţiile şi suprafeţele utilizator necesită adaptarea driverelor, conversii şi comunicaţii de date. La planificarea unei aplicaţii distribuite este hotărîtoare numirea unui angajat propriu care să răspundă de produsul final. Acesta trebuie să fie cineva care să supravegheze activităţile efectuate cu datele. El trebuie să aibă o privire de ansamblu asupra drepturilor de acces la date şi este răspunzător pentru operaţiile de salvare (backup) şi restaurare (recovery) în caz de necesitate. În domeniul lui de activitate intră şi administrarea structurii bazei de date şi măsurile de acordare cu DBMS-ul (Data Basis Management System) cu care a fost dezvoltată aplicaţia. Această persoană, numită şi administratorul bazei de date (DBA - Data Basis Administrator), joacă un rol decisiv atunci cînd distribuirea prelucrărilor se vrea să fie un succes. Ideea de a crea o instanţă independentă de administrare şi control este străină celor mai mulţi programatori, dar elaborarea unor aplicaţii distribuite este ineficientă fără un DBA. Dar pentru că multe drumuri duc la Roma, punctele decisive în cazul unei aplicaţii distribuite sînt designul şi posibilitatea de restructurare. Din acest motiv este hotărîtor ca proiectantul să cunoască exact cerinţele de specialitate impuse de o bază de date distribuită. La analiză două puncte stau în fruntea listei de probleme:

- care date să fie stocate local şi care global
- cum trebuie legate componentele între ele.

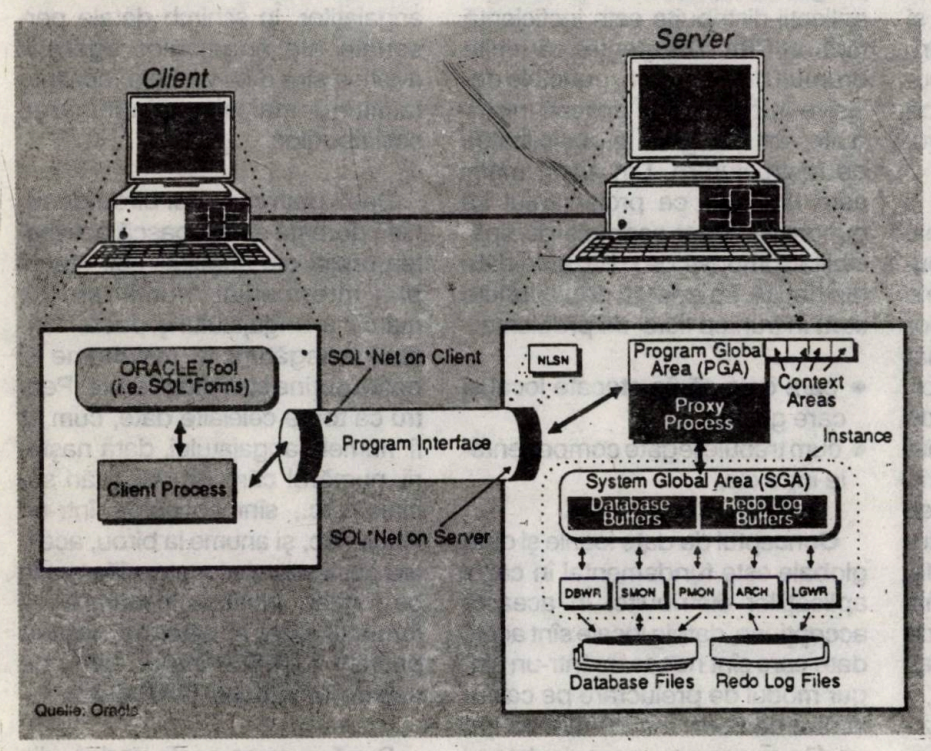
Conceptul de date locale şi date globale este fundamental în cazul aplicaţiilor distribuite. În această accepţiune, datele locale sînt acele date care sînt necesare într-un singur modul de prelucrare pe calculatorul de la un loc de muncă, pe un file-server sau pe calculatorul unei secţii. Datele globale, care de cele mai multe ori sînt decisive pentru rularea corectă a unei aplicaţii, sînt date care sînt prelucrate de două module aflate pe acelaşi nivel ierarhic, ambele fiind legate de treapta de prelucrare superioară. Datele globale pot fi memorate pe un file-server şi folosite de două staţii de lucru din reţea, sau pot fi memorate pe calculatorul unei secţii care deserveşte două servere de baze de date din reţea. Un exemplu va clarifica diferenţa dintre datele globale şi cele locale şi importanţa cunoaşterii, încă din faza de analiză, a punctelor de vedere locale şi globale.

Să presupunem că o agenţie de aviaţie îşi planifică un sistem contabil, care să poată fi utilizat gratuit de către agenţiile independente de turism pentru rezervarea de locuri şi că în plus ea doreşte să-şi dezvolte un sistem de evidenţă a personalului şi de calculul salariilor. La analiză se va ajunge la concluzia că majoritatea datelor personale ale angajaţilor din birourile de turism vor trebui stocate şi ţinute la îndemînă local. Singurele date personale care trebuie stocate global sînt numerele de înmatriculare ale angajaţilor. În schimb datele personale ale angajaţilor agenţiei aviatice sînt date globale, aceasta facilitînd mai ales planificarea schimburilor.

Dacă patronul unui birou de turism doreşte să cunoască activitatea unuia din angajaţii săi, atunci prin intermediul "numărului de marcă" al angajatului îşi poate construi o legătură la mainframe şi poate obţine statistica cerută. Pentru că toate celelalte date, cum ar fi: numele angajatului, data naşterii, numărul cărţii de asigurări sociale, etc., sînt necesare într-un singur loc, şi anume la birou, acestea pot fi stocate local, indiferent în ce format. Aceasta înseamnă că formatul local al datei de naştere poate fi "LLAAZZ" într-un birou de turism, iar într-altul "AAAAZZLL".

După cum se poate vedea, din acest exemplu simplu, este necesară o analiză preliminară detaliată înainte de a putea spune despre o dată că este locală sau globală. Dacă majoritatea datelor sînt declarate locale, atunci trebuie să existe mai multe programe "locale" pentru administrarea acestor date şi pentru a putea obţine rapoarte despre ele, în acelaşi timp însă vor creşte cheltuielile de dezvoltare şi de întreţinere a sistemului. În exemplul dat anterior calculul salariilor se face în birourile locale de turism. Necesară este doar stabilirea unui interval de numere între care să se dea numerele de înmatriculare personale pentru fiecare birou de turism.

Aspectele analizei datelor locale şi globale poate depăşi limitele organizatorice. De exemplu într-un sistem de rezervare şi control al biletelor de avion pot exista şi agenţii cu birouri în toată lumea. Fiecare agent ar putea, în acest caz, să hotărască liber, deci local, asupra unui segment al contingentului de rezervări. Modulele care se referă la controlul zborurilor ar trebui să ofere informaţii continue, actualizate, despre fiecare aeroport în parte, deci trebuie să aibă acces la date globale.



Arhitectură multi-proces în cazul unui server Oracle

Cel de-al doilea criteriu important, de aceeaşi importanţă ca şi definirea datelor locale şi globale, este stabilirea modului de interconectare a componentelor. Cu toate că accentul în acest caz cade pe programe şi nu pe date, şi în acest caz cel mai bun coordonator este administratorul bazei de date, pentru că el răspunde, printre altele, de design-ul bazei de date. Cerinţele unei aplicaţii distribuite seamănă mai mult cu cele ale unui magazin universal, decît cu cerinţele tradiţionale ale unei aplicaţii, deoarece trebuie să se ţină seama de puterea diferită de calcul a sistemelor şi de multitudinea cerinţelor individuale pe care trebuie să le satisfacă programul. Rezultatul este că vor trebui dezvoltate multe bucăţi mici de programe dificile.

De multe ori diferenţele dintre programe constau doar în faptul că introducerea datelor pentru cîmpuri identice se face din poziţii de pe ecran diferite. Mulţi programatori cu experienţă sînt tentaţi să ocolească aceste diferenţe şi să încerce să dezvolte module multifuncţionale. Aceasta poate duce la obţinerea unor module atît de inflexibile încît ele nu vor mai putea fi adaptate la situaţii noi şi vor trebui rescrise. Scopul dezvoltării unei aplicaţii distribuite este de a reduce fiecare cerinţă la componentele ei elementare. Fiecare componentă trebuie să conţină numai acele instrucţiuni de care are nevoie în timpul lucrului.

Această tehnică de atomizare, care este de fapt o analiză top-down, s-a răspîndit la toate nivelele programării, începînd de la scrierea de rutine pentru module, construirea programelor din module şi pînă la integrarea programelor într-o aplicaţie completă.

Cuplarea diferitelor componente într-o aplicaţie se poate face în variante treptate. La o aplicaţie cu legături slabe fiecare componentă este un atom, şi numai funcţiile care comandă acea componentă sînt legate cu aplicaţia. O aplicaţie cu legături strînse se caracterizează prin aceea că diferitele componente sînt grupate în mod logic şi funcţiile care acţionează asupra componentelor sînt asociate cu grupele care le supraordonează. Această metodă de design reduce puternic flexibilitatea şi capacitatea de întreţinere a unei aplicaţii. Dacă de exemplu funcţiunea de rezervare de locuri ar fi strîns legată cu orarul de control al zborurilor, atunci programele aplicaţiei ar fi prea strîns legate între ele. Cu toate că această combinaţie ar reduce necesitatea proiectării şi codificării diferite a măştilor de ecran, aplicaţia ar putea funcţiona doar atunci cînd derularea procesului din ambele domenii ar fi absolut identică tastă cu tastă.

Dacă cerinţele asupra unei rezervări s-ar schimba peste noapte, ar trebui verificate şi reproiectate părţi ale aplicaţiei, problemă care se iveşte întotdeauna cînd componentele aplicaţiei sînt prea strîns legate între ele. În plus exemplul aproape că depăşeşte deja imaginaţia unui programator prevăzător. Dar practica este de regulă încă mult mai complexă decît teoria.

Alte puncte de vedere de care trebuie să ţinem cont încă din faza de analiză sînt competenţa operaţională şi funcţională ale unui utilizator. Ne referim aici la dreptul de acces la anumite date, la stabilirea grupei de utilizatori, felul în care se va face manipularea datelor permise unei anumite peroane, etc. Este foarte important ca aceste lucruri să fie stabilite din timp.

Pe lîngă aceste aspecte ale protecţiei datelor, de mare importanţă este şi verificarea corectitudinii datelor încărcate. Aici nu ne referim doar la încadrarea valorilor cîmpurilor între anumite limite. Multe aplicaţii sînt prevăzute cu cîmpuri de text vide (zone de rezervă). Aşa se poate întîmpla ca după un timp în aceste cîmpuri să se adune informaţii total diferite. Astfel într-un birou cîmpul de rezervă poate conţine stadiul datoriilor din ultima seară de popice, în timp ce în alt birou acelaşi cîmp poate conţine date despre locul de parcare al angajatului în parcarea firmei.

Diferenţa hotărîtoare între designul unei aplicaţii tradiţionale şi cel al unei aplicaţii distribuite este gradul mult mai ridicat de complexitate. Într-un sistem tradiţional programele şi datele se află de regulă într-un singur loc şi acolo sînt şi folosite. Un sistem distribuit se face remarcat prin aceea că programele şi datele sînt memorate în locuri diferite şi rolul programelor poate varia în funcţie de aplicaţia care tocmai se află în execuţie.

Cu gradul de complexitate creşte şi capacitatea. Dacă aspectele de: integritate a datelor, controlul accesului simultan la date, tratarea erorilor, operaţiile de back-up şi recovery sînt tratate ordonat, atunci o aplicaţie distribuită poate aduce avantaje semnificative în comparaţie cu un sistem tradiţional. O condiţie este, ca de exemplu, accesul la datele globale să fie cît mai redus. Dacă se ţine cont de această indicaţie, atunci utilizatorul poate lucra cu datele sale locale şi atunci cînd serverul sau calculatorul mare nu este operaţional. Acesta este un avantaj mare faţă de sistemele centralizate, pentru că la acestea dacă "pică" server-ul, toate lucrările stau.

Aspectele centrale ale unei aplicaţii distribuite sînt integritatea datelor şi controlul accesului simultan asupra datelor. Există o serie de tehnici, pe care programatorul le poate folosi, pentru a proteja datele înainte, în timpul sau după prelucrarea propriu-zisă. Integritatea datelor cuprinde: exactitatea, valabilitatea şi protecţia datelor. Într-o aplicaţie distribuită integritatea datelor este asigurată de către programe. Într-un sistem de rezervare de locuri la avion, agenţii din mai multe oraşe se ocupă de rezervarea locurilor pentru aceeaşi cursă. Pentru a asigura integritatea datelor trebuie găsită o metodă care să aibă grijă ca acelaşi loc să nu fie rezervat decît o singură dată. Din motive de siguranţă, la anumite date, unii utilizatori trebuie să aibă acces numai în citire. Modificări asupra datelor se vor permite numai anumitor grupe de utilizatori. Controlul acceselor simultane la date se ocupă cu modul în care aplicaţia tratează cererile de interogare. Problemele serioase puse de accesul simultan la date pot fi depistate numai cu tehnici de proiectare încercate. Un domeniu de probleme este ridicat de succesiunea în care resursele aplicaţiei, cum ar fi: timp CPU, loc în memorie sau pe harddisk, imprimantă, sau chiar înregistrările individuale ale unei tabele, pot fi solicitate.

În timpul folosirii acestor resurse de lucru se poate ajunge la un "deadlock" - un blocaj total, reciproc. Un astfel de blocaj poate să apară, de exemplu, în cazul în care un program aşteaptă eliberarea unei resurse folosite de un alt program, iar acest al doilea program, aşteaptă o resursă acordată primului program.

O altă problemă este cunoscută sub numele de "lock out" - închis pe dinafară. Această situaţie poate să apară atunci cînd un program foloseşte în mod exclusiv o resursă, care este necesară şi unui alt program, care îşi poate continua execuţia abia după accesul la această resursă. Prin această excludere nu numai că sistemul va merge mult mai încet, dar sistemul poate şi "claca". Dacă un angajat al unui birou de turism îşi propune o rezervare, dar lasă tranzacţia nenterminată şi merge să ia masa de prînz, atunci ne putem închipui că tranzacţia blochează resurse critice şi că în acest timp nu se pot face alte rezervări de locuri.

Există tabele care oferă protecţie împotriva fenomenelor de "deadlock" şi "lock out", ele acordînd programelor un timp maxim de prelucrare pentru o anumită resursă. Dacă acest timp este depăşit şi dacă nu urmează o altă cerere semnificativă, resursa este eliberată. Ca exemplu dăm situaţia rezervării de loc neterminate. Prin folosirea unei tabele de alocare a resurselor, la început tranzacţia va fi întreruptă. În pasul următor trebuie să ne îngrijim ca starea anterioară să fie refăcută, iar la urmă trebuie să redăm folosirii generale toate resursele care au fost blocate prin tranzacţie.

Design-ului îi urmează programarea. Există mai multe modalităţi de a dezvolta o aplicaţie, variantele diferenţiindu-se în funcţie de mediul de dezvoltare şi instrumentele alese. Două probleme fundamentale, care joacă momentan un rol important în discuţiile despre bazele de date distribuite sînt: arhitectura client-server şi bazele de date bazate pe SQL. În cadrul dezvoltării aplicaţiilor distribuite modelul client-server prezintă cel mai mare interes. în acest caz programele se află pe calculatorul "client" (de obicei un PC), legat într-o reţea LAN cu "server"-ul bazei de date. Server-ul prelucrează interogările clientului, pregăteşte rezultatele şi răspunde cererii.

Există trei modele de bază: - clienţi inteligenţi şi serveri proşti - clienţi proşti şi serveri inteligenţi - clienţi inteligenţi şi serveri inteligenţi

În cazul clienţilor inteligenţi şi a sever-ilor proşti, pregătirea datelor şi interogările au loc în întregime pe calculatorul client, server-ul fiind folosit doar pentru introducerea şi extragerea datelor. Această variantă este foarte răspîndită la mul-

te sisteme de management a bazelor de date de tipul "Low-End-PC". Terminalele proaste şi server-ele inteligente acoperă celălalt capăt al ierarhiei; această constelaţie domină în cazul sistemelor centralizate cu mainframe-uri sau minicalculatoare. În cazul în care atît server-ele cît şi clienţii sînt inteligenţi, datele sînt prelucrate cu o integritate îmbunătăţită şi resursele reţelei sînt exploatate mai eficient.

Avantajele ultimului model numit pot fi exemplificate printr-o interogare simplă ad-hoc. Un angajat al unei firme de turism doreşte să confirme cererea de rezervare a unui loc pentru domnul Leon Moraru şi introduce numele "Moraru" în cîmpul "Nume" din macheta de pe ecran, ca fiind singurul criteriu de căutare.

În cazul unui client inteligent şi a unui server prost, server-ul trebuie să-i trimită clientului întregul fişier de rezervări, iar acesta trebuie să caute singur înregistrarea respectivă. În varianta cu un client prost şi un server inteligent, server-ul ar trimite clientului toate înregistrările care conţin valoarea "Moraru" în cîmpul "Nume". Numele "Moraru" fiind destul de frecvent, pe ecran este posibil să apară destul de multe înregistrări. Interogarea ar trebui detaliată specificîd, de exemplu, şi prenumele domnului "Moraru". În acest caz însă înregistrările transmise deja clientului devin fără valoare şi căutarea trebuie reluată de la început.

În cazul clienţilor inteligenţi şi a server-elor inteligente, înregistrările deja selectate nu trebuie "aruncate", deoarece atît clientul cît şi server-ul ştiu că ele sînt deja afişate la client. La o detaliere a interogării, clientul va produce un subşir de înregistrări, din şirul obţinut de la prima interogare, subşir care va conţine valoarea "Leon" în cîmpul "Prenume".

O altă tendinţă în dezvoltarea aplicaţiilor cu baze de date este folosirea SQL-ului pentru accesul la baza de date, pentru interogarea bazei de date şi pentru manipularea datelor. SQL este un limbaj neprocedural, dezvoltat pentru bazele de date relaţionale, care îşi are rădăcinile în teoria mulţimilor din matematică. SQL a fost obţinut în urma unui proiect de cercetare al IBM-ului. Limbajul constă din trei componente esenţiale: limbajul de definire a datelor (DDL, Data - Definition-Language), limbajul de control a datelor (DCL, Data - Control-Language) şi limbajul de manipulare a datelor (DML, Data-Manipulation-Language).

DDL oferă mecanisme pentru crearea şi modificarea structurilor bazei de date. DCL se ocupă de supravegherea valabilităţii datelor dintr-un mediu multi-user. Componenta cea mai utilizată este DML, componentă care oferă funcţiunile de interogare, adăugare şi modificare de date.

Deoarece SQL este recunoscut în lumea calculatoarelor ca un standard general, el a devenit fundamentul multor aplicaţii cu baze de date distribuite. Cu ajutorul lui programele pot utiliza în comun datele aflate în reţele diferite şi sub sisteme de operare de reţea diferite. Producătorii de soft au implementat limbajul SQL într-o multitudine de produse, începînd cu mediul de programare 4GL, cu limbajele de programare şi pînă şi în pachetele de soft orientate spre utilizator cum ar fi Lotus 1-2-3. Produsele bazate pe SQL au devenit un element de bază pentru dezvoltarea aplicaţiilor distribuite, chiar dacă SQL presupune echipamente hard de înaltă calitate, şi viteza de lucru pe un PC mai lasă încă de dorit. Există o multitudine de instrumente pentru dezvoltarea aplicaţiilor distribuite, baze de date distribuite, servere de baze de date, medii de dezvoltare şi alte utilitare. O bază de date distribuită poate funcţiona concomitent pe platforme hard diferite şi pe sisteme diferite dacă s-a realizat comunicaţia şi compatibilitatea între platforme.

O bază de date este susţinută, de regulă, de SQL, de un mediu de dezvoltare 4GL şi de mai multe limbaje de programare. În continuare vor fi prezentate pe scurt două exemple de baze de date distribuite: "Oracle" şi "Informix". Alte nume de baze de date care ar putea fi enumerate în acest context ar fi: "Empress", "Ingres", "Sybase" sau "Zim".

"Oracle" a oferit prima implementare completă a lui SQL într-o bază de date distribuită deja în 1979. Pentru susţinerea limbajelor de programare C, Cobol sau Fortran ca şi Front-Ends (comunicaţie cu exteriorul bazei), Oracle oferă în plus instrumente Front-Ends proprii:

- SQL Forms, un instrument de dezvoltare interactiv
- SQL Plus, un instrument interactiv pentru cereri şi liste
- SQL Menu, un instrument pentru proiectarea interactivă a meniurilor
- SQL Star, un instrument pentru obţinerea de cod independent de protocol
- SQL Report Writer, un instrument interactiv pentru liste, şi
- SQL Graph, un instrument interactiv pentru grafică.

Informix oferă facilităţi pentru prelucrarea datelor binare, a "regulilor" care trebuie folosite în cadrul înregistrărilor selectate ale unei baze de date şi mecanisme pentru ridicarea eficienţei operaţiilor I/O. Informix se sprijină pe mai multe instrumente de dezvoltare pentru PC-uri, pe o interfaţă pentru limbajul de programare C, pe pachetul propriu de programe integrate Smart Ware şi pe un Add-in Lotus 1-2-3 pentru interogarea datelor.

Server-ele bazelor de date, o altă categorie de instrumente pentru dezvoltarea de aplicaţii distribuite, se bazează de regulă pe modelul client-server. Cele mai

multe servere de baze de date constau din două componente: partea de "Back-End", care se află pe serverul reţelei şi răspunde de integritatea datelor şi expedierea răspunsurilor, şi partea de "Front-End", care se află pe PC-ul utilizatorului, căruia îi oferă posibilitatea să interogheze baza de date şi să reprezinte rezultatele obţinute în diferite formate. Se pot deosebi două categorii de servere de baze de date: servere de baze de date pe bază PC şi servere de baze de date distribuite. Prima categorie se limitează la sistemele care au fost legate în reţele cu sistemul de operare MS-DOS sau OS/2, în timp ce a doua categorie poate lucra atît cu sisteme de operare de reţea, cît şi cu sisteme de operare multiutilizator, cum ar fi Unix sau MVS.

Majoritatea serverelor de baze de date aparţin categoriei de servere bazate pe PC-uri. Aşa sînt, de exemplu, produsele "SQL Base" sau "SQL-Server". Integrarea instrumentelor "Front-End" se face prin intermediul interfeţei de programare "DB-Library" a server-ului SQL. SQL Base dispune de un mediu de proiectare grafic, care se bazează pe Microsoft Windows şi pe Presentation Manager. Facilităţile grafice pot fi utilizate pentru proiectarea de aplicaţii, care să poată accesa baze de date SQL Base.

Din categoria server-elor de baze de date distribuite pot fi enumerate "Netware SQL" şi server-ul "Oracle". Netware SQL oferă facilităţi pentru servere de baze de date pe platforme DOS, OS/2, Macintosh şi VAX/VMS. Novell se concentrează pe dezvoltarea de suporţi pentru partea de "Back-End". Partea "Front-End" este cedată caselor independente de soft. Oracle a ales o altă cale şi oferă singură toate instrumentele necesare dezvoltării unei aplicaţii. Astfel Oracle poate oferi un produs a cărui suprafaţă utilizator este identică pe toate platformele, de la DOS, Unix, DEC VMS şi pînă la MVS al IBM-ului.

O altă categorie de instrumente de dezvoltare o constituie, aşa numit-a, "Developers Workbench", asemănătoare cu bancul de lucru al unui tîmplar. La fel cum bancul de lucru al acestuia conţine toate sculele de care ai nevoie pentru construirea unei piese de mobilier, şi Workbench conţine toate instrumentele necesare pentru construirea unei aplicaţii distribuite. Ofertanţi de astfel de colecţii de instrumente sînt Novell, Hewlett-Packard şi Santa Cruz Operation.

Workbench-ul pentru programatori al lui Novell, care conţine un kit de dezvoltare software, este un set de instrumente pentru scrierea de module încărcabile în reţea (Netware Loadable Modules, NLM). Un proiectant poate scrie procese client-server, în forma NLM, care să poată fi rulate sub sistemul de operare de reţea Netware 386. Amintim şi setul de instrumente "Remote Procedure Call" al firmei Netwise. El conţine funcţii care se ocupă de comunicarea interproces între programe care sînt rulate pe calculatoare diferite. Printre protocoalele de transport suportate de RPC se numără şi TCP/IP, protocolul OSI şi DECnet. RPC poate lucra pe sistemele de operare de reţea Netware, LAN Manager, şi altele. Cu instrumentele RPC, proiectantul trebuie să stabilească, mai întîi, care aplicaţii vor fi executate "de la distanţă" şi care "local". Urmează apoi o dezvoltare a programului ca o aplicaţie de sine stătătoare, dar construită modular, pentru a se simplifica procesul de programare. Codul programului va fi scris într-un limbaj standard, cum ar fi C, folosindu-se însă extensiile de limbaj oferite de RPC. În fine, aplicaţia va fi compilată şi link-editată. În execuţie RPC interpretează extensiile de limbaj, folosind instrumentul ca pe un fel de preprocesor al aplicaţiei care generează cod standard în C.

Un alt workbench pentru dezvoltarea de aplicaţii distribuite este NCS (Network Computing System) al firmei Hewlett Packard. El conţine un compilator, un "Location Broker" şi funcţii pentru dezvoltarea de RPC. Sînt susţinute limbajele de programare C şi Pascal, sub sistemele de operare HP/UX, VAX/VMS, Ultrix şi Sun OS. Produsul "Open Desktop" al firmei SCO este un software care conţine sistemul de operare SCO Unix, suprafaţa utilizator X-Windows, baza de date Ingres şi alte componente de dezvoltare. În sfirşit, la rubrica unelte de dezvoltare există şi grupa de "diverse". În aceasta intră, de exemplu, produse ca "Network File System" (NFS) al firmei Sun, un sistem de gestiune a fişierelor distribuite, care permite accesul la date aflate pe mai multe platforme de sisteme de operare. NFS susţine mai multe tipuri de echipamente hard, sisteme soft, medii de dezvoltare, şi în fine limbajele de programare şi compilatoarele existente pe acestea.

Odată ce aplicaţiile distribuite vor fi acceptate în întreprinderi, producătorii vor oferi şi mai multe instrumente de dezvoltare a aplicaţiilor. În acest context ne putem imagina deja un utilitar care să îmbunătăţească comunicarea între procese. Proiectanţii aşteaptă deja o ofertă mai mare de rutine şi biblioteci de programe pentru comunicaţia low-level.

Tendinţa de a dezvolta aplicaţii distribuite influenţează şi modul de organizare al secţiilor MIS. Multe întreprinderi şi-au creat deja noi centre informatice, şi noi locuri de muncă, care să fie răspunzătoare pentru realizarea de aplicaţii distribuite. În sarcina lor intră şi instruirea personalului care va intra în contact cu astfel de aplicaţii. Dacă vrem să stăpînim problemele legate de protecţia şi integritatea datelor, de accesul simultan la date, atunci crearea de astfel de echipe de lucru este o necesitate. Acest articol nu face decît să arunce o scurtă privire în aspectele centrale ale aplicaţiilor distribuite.

(Computer Persönlich 22/90, Steve Guengerich, Dwight Williams)

(continuare în pag. 37)

Se poate formata şi altfel...

Pe un AT cu o unitate floppy de tip MF, utilizatorul nu poate produce pe discbete simple (2S,DD) dischete de 420 kbyte, 720 kbyte sau 800 kbyte. O alternativă o oferă programul "mformat".

Este o dilemă chinuitoare: să te mulţumeşti cu dischetele simple şi ieftine, însă cu capacitatea rizibilă de 360 kbyte sau să utilizezi dischetele încă destul de scumpe de mare densitate (HD) cu capacitatea de 1,2 Mbyte. Căci formatele suplimentare posibile nu sînt sprijinite de MS-DOS nici pentru unităţile floppy MF de 5,25", nici pentru unităţile noi de 3,5". Aceste formate suplimentare sînt: simplă densitate, 42 piste, 10 sectoare/pistă = 420 kbyte; 80 piste, 9 sectoare/pistă = 720 kbyte şi 80 piste, 10 sectoare/pistă = 800 kbyte.

Pentru a formata însă o dischetă cu 720 kbyte sau 800 kbyte trebuiesc îndeplinite două condiţii

1) ROM-BIOS-ul trebuie să fie de versiune 3.xx sau mai mare (ca să cunoască şi noile unităţi de 3,5") şi

2) MS-/PC-DOS-ul trebuie să fie cel puţin versiunea 3.2

Numai atunci programul care urmează poate aduce fără limitări dischetele la formatul dorit.

Structura informaţiilor de pe o dischetă este următoarea:

- bootsector
- primul FAT (tabela de alocare a fişierelor)
- una sau mai multe copii FAT
- directorul principal sau rădăcină (root)
- zona de date

Versiunile mai vechi de DOS (înainte de 3.2) au folosit pentru recunoaşterea formatului unei dischete primul octet din primul FAT, aşa-numitul media-byte.

Tabela semnificaţie media-byte					
byte:	**tip:**	**feţe**	**piste**	**sect /pistă**	**capacitate(kbyte)**
f0h	3,5"	2	80	18	1440
f8h	harddisk				
f9h	5,25"	2	80	15	1200
	3,5"	2	80	9	720
fah	5,25"	1	80	8	320
	3,5"	1	80	8	320
fbh	5,25"	2	80	8	640
	3,5"	2	80	8	640
fch	5,25"	1	40	9	180
fdh	5,25"	2	40	9	360
feh	5,25"	1	40	8	160
ffh	5,25"	2	40	8	320

Pe baza acestui media-byte DOS şi ROM-BIOS puteau recunoaşte tipul dischetei.

După cum se poate vedea din tabelă, media-byte-ul nu mai era suficient după apariţia unităţilor de 3,5", deoarece o aceeaşi valoare a media-byte-ului descrie formate total distincte pe dischete de 3,5" şi pe dischete de 5,25". Trebuia găsită deci o nouă soluţie pentru deosebirea formatului.

Versiunile DOS de la 3.2 încoace ne oferă o cu totul altă modalitate de recunoaştere a formatului.

Şi anume ele se pot acorda singure pe formatul dischetei cu care se lucrează, exploatînd informaţiile din blocul parametrilor discului din cadrul boot-sectorului dischetei DOS.

Acest bloc al parametrilor discului se află în boot-sectorul dischetei, deci în primul sector fizic.

Boot-sectorul dischetei

Adresa	Conţinut	Tip
+00h	instrucţiune de salt la rutina de boot	3 byte
+03h	numele producătorului şi număr de versiune	8 byte
+0bh	bytes/sector	1 word
+0dh	sectoare/cluster	1 byte
+0eh	numărul sectoarelor rezervate (sectoare inaintea primului FAT)	1 word
+10h	numărul FAT-urilor	1 byte
+11h	numărul maxim al intrărilor în directorul rădăcină	1 word
+13h	numărul sectoarelor de pe dischetă	1 word
+15h	media-descriptor (ca şi primul octet din FAT)	1 byte
+16h	numărul de sectoare per FAT	1 word
+18h	sectoare/pistă	1 word
+1ah	numărul capetelor de scriere/citire	1 word
+1ch	numărul sectoarelor ascunse (distanţa pînă la primul sector)	1 word
+1eh-1ffh	rutina de boot	

De-abia această metodă permite recunoaşterea diferitelor formate de dischetă.

Formatarea unei dischete la 720 kbyte într-o unitate MF (care în mod normal lucrează cu dischete de capacitate mare) este relativ simplă. Sistemului de operare trebuie numai să i se "explice" că este vorba de o dischetă de 3,5" formatată la 720 kbyte. Deoarece DOS, după cum s-a arătat mai sus, face recunoaşterea formatului prin intermediul boot-sectorului, e imediată ideea modificării acestuia în aşa fel încît DOS să recunoască discheta ca una de 3,5" formatată la 720 kbyte.

Dacă o dischetă ar fi formatată în simplă densitate cu 80 piste, 9 sectoare/pistă, modificarea boot-sectorului ar fi suficientă ca discheta să fie recunoscută de DOS ca dischetă de 720 kbyte.

Dar cum se poate formata o dischetă în simplă densitate cu 80 piste?

ROM-BIOS disponibilizează o rutină cu ajutorul căreia se poate modifica cîte o pistă a unei dischete (funcţia 5, Int 19).

Dacă se încearcă formatarea dischetelor de simplă densitate cu ajutorul acestei rutine la 80 de piste, capul de citire/scriere al unităţii va ajunge în poziţia limită finală deja de pe la pista 42. Aceasta nu se datorează faptului că unitatea poate formata numai 40 de piste în simplă densitate, ci ROM-BIOS-ului, care la formatarea în simplă densitate face totdeauna un dublu pas la trecerea la pista următoare. Dacă acest lucru ar putea fi împiedicat, nu ar mai exista nici o stavilă în calea formatării la 80 de piste.

ROM-BIOS-ul "ţine minte" dacă trebuie executat un dublu sau simplu pas la trecerea la pista următoare.

Modificarea octetului de memorie aflat la adresa 0040:0090 la valoarea 54h are ca efect executarea unui pas simplu la trecerea la pista următoare.

Adresarea directă a acestei celule de memorie este de altfel singura porţiune dependentă de hard a acestui program. Nu trebuie ascuns faptul că la alte calculatoare, unde comutarea de la simplu pas la dublu pas se face prin intermediul altei celule de memorie, programul nu va putea funcţiona în forma prezentată. Dacă însă este cunoscut octetul cu pricina, modificări minore în program pot duce la reuşită.

După formatare nu mai trebuie apoi scrise pe dischetă decît boot-sectorul cu blocul parametrilor disc, sectoarele FAT şi sectoarele directorului rădăcină eventual cu un nume pentru volum. Dacă discheta urmează să fie folosită ca dischetă sistem pentru încărcare, la terminarea programului ar trebui utilizată comanda DOS "sys a:", pentru a transpune fişierele sistem pe dischetă.

Programul se startează la nivel DOS cu comanda "MFORMAT". După scurt timp pe ecran apare meniul principal:

Meniul principal MultiFormat pe unitati de 5,25"
if Freeware
Sint disponibile urmatoarele optiuni:
(sus-jos alege, Enter starteaza)
Unitatea : A
Verificare: da
360 kbyte
420 kbyte
720 kbyte
800 kbyte
1.2 Mbyte
1.4 Mbyte
Terminare

Cu ajutorul tastelor cursor (sus-jos), bara colorată poate selecta punctul de meniu dorit. O apăsare pe tasta RETURN determină selecţia punctului de meniu respectiv. Se poate lucra după dorinţă pe discheta A: sau B:, cu sau fără verificare. Din motive de siguranţă, se recomandă să se lucreze cu opţiunea de verificare, pentru a asigura o mai mare securitate a datelor. Dacă există însă certitudinea că discheta este complet amente fără erori, se poate renunţa la verificare, accelerînd astfel formatarea. Porţiunea deja formatată a discului este afişată de acest program sub formă de procente.

La terminarea procesului de formatare, programul cere un nume pe care să-l primească discheta. La introducerea unui nume, programul stochează pe dischetă odată cu acesta şi data şi ora curentă. Dacă discheta urmează să fie folosită ca şi dischetă sistem, în acest loc nu trebuie dat un nume ci doar acţionată tasta RETURN. După terminarea programului de formatare se pun şi fişierele sistem pe dischetă, cu comanda "sys".

Programul a fost astfel conceput încît o modificare a formatelor de dischetă este uşor de operat. De exemplu, se pot modifica uşor numărul de piste respectiv al numărului de sectoare/pistă în apelul rutinelor corespunzătoare de formatare.

Fără o modificare suplimentară în program însă nu se pot formata decît cel mult 10 sectoare/pistă. Dacă se doreşte o formatare cu mai mult de 10 sectoare/pistă, mai trebuie modificată şi tabela de parametri ai unităţii. Această tabelă de 11 octeţi este adresată

Tabela de parametri ai unităţii

Octet	Funcţie
0	timpul maximal necesitat de capul de citire scriere pentru a trece de la o pistă la alta (biţii 0-3); timpul de descărcare al capului (biţii 4-7)
1	bit 0: 1- controller-ul unităţii foloseşte DMA; biţii 2-7: timpul de încărcare al capului
2	specifică timpul cît motorul unităţii rămîne încă activ după efectuarea unei operaţii, folosind un tact de timer de 18,2 impulsuri/secundă. O valoare de 36 de exemplu determină un timp de cca. 2s pînă la deselectarea motorului
3	numărul de octeţi/sector 0 - 128 octeţi 1 - 256 octeţi 2 - 512 octeţi 3 - 1024 octeţi pentru dischete şi unităţi uzuale se foloseşte valoarea 2.
4	numărul maxim de sectoare/pistă
5	intervalul logic (gap) pînă la următorul sector; defineşte un interval care trebuie să treacă înainte ca următorul sector şă poată fi accesat sub capul de citire/scriere.
6	lungimea transferului de date; defineşte lungimea maxima a unui transfer pentru cazurile în care acesta nu e specificat în comandă.
7	intervalul fizic (gap la formatare) pînă la următorul sector; trebuie să fie totdeauna mai mare decît intervalul logic!
8	caracterul cu care sînt iniţializaţi toţi octeţii de date în cursul formatării; deobicei, 0f6h.
9	defineşte timpul de stabilire al capului după o schimbare de poziţie, în unităţi de 1 ms; depinde de tipul unităţii de dischete.
10	intervalul care trebuie să treacă înainte ca motorul să ajungă la turaţia nominală, în unităţi de 1/8 secunde.

de Interrupt 1eh. Pointerul către această tabelă se găseşte în memorie la adresa 0000:0078. în ea se găsesc parametri controller-ului de floppy asociat.

Dacă se doreşte o formatare cu mai mult de 9 sectoare/pistă, trebuie modificate intervalul logic respectiv fizic între sectoare specificat în această tabelă. Valori exacte nu pot fi specificate. Găsirea valorilor potrivite se poate face numai experimentînd.

Dischetele formatate în acest fel pot fi desigur copiate cu ajutorul programului "DISKCOPY". în acest caz însă totdeauna şi discheta ţintă trebuie în prealabil formatată la fel; altfel programul se abortează cu un mesaj de eroare.

Indicaţie: cei interesaţi în acest program pot comanda o dischetă HD noname, nouă, cu programul pe ea, la preţul de 300 lei, de la redacţia revistei.

(Computer Persönlich 26/89,
Volker Bührmann,; Iosif Fettich);

Listing-ul programului Mformat;

```
program mformat
(* if - freeware *)

(* cu afisarea capacitatii formatate in procente *)

uses crt, dos;
const
        maxwrite  = 12;  (* lungimea maxima a mesajelor *)
        retries = 10;    (* 10 incercari la erori de citire *)
        lng =2; (* lungimea unui sector - totdeauna 512 octeti *)
        bootsec:
                array[1..354] of byte =
($eb,$3e,$90,$4d,$2e,$46,$4f,$52,$4d,$41,$54,$00,lng,$00,$01,$00,

$02,$70,$00,$d0,$02,$fd,$02,$00,$09,$00,$02,$00,$00,$00,$00,$00,

$00,$00,$00,$00,$0f,$00,$00,$00,$00,$01,$00,$00,$00,$00,$00,$00,

$00,$00,$06,$00,$01,$04,$0c,$00,$00,$00,$00,$00,$00,$00,$00,$00,

$fa,$fc,$33,$c0,$8e,$d0,$bc,$00,$7c,$36,$c5,$36,$78,$00,$1e,$56,

$8e,$c0,$8d,$3e,$20,$7c,$b9,$0b,$00,$ac,$26,$80,$3d,$00,$74,$03,

$26,$8a,$05,$aa,$e2,$f3,$33,$c0,$8e,$d8,$a3,$7a,$00,$c7,$06,$78,

$00,$20,$7c,$fb,$cd,$13,$73,$03,$e9,$b4,$00,$8a,$2e,$30,$7c,$8a,

$0e,$32,$7c,$8a,$36,$31,$7c,$b2,$00,$bb,$00,$05,$b8,$01,$02,$cd,

$13,$72,$e5,$8d,$36,$62,$7d,$8b,$fb,$b9,$0b,$00,$f3,$a6,$75,$0d,

$bf,$20,$05,$b9,$0b,$00,$f3,$a6,$74,$1a,$e9,$89,$00,$8d,$36,$78,

$7d,$8b,$fb,$b9,$0b,$00,$f3,$a6,$75,$7c,$bf,$20,$05,$b9,$0b,$00,

$f3,$a6,$75,$72,$33,$d2,$a1,$1c,$05,$f7,$36,$0b,$7c,$fe,$c0,$a2,

$38,$7c,$bb,$00,$07,$8a,$2e,$33,$7c,$8a,$0e,$35,$7c,$8a,$36,$34,

$7c,$b2,$00,$a1,$18,$7c,$2a,$c1,$fe,$c0,$50,$b4,$02,$cd,$13,$58,

$72,$3d,$28,$06,$38,$7c,$76,$1a,$b4,$00,$52,$f7,$26,$0b,$7c,$5a,

$03,$d8,$b1,$01,$fe,$c6,$3a,$36,$1a,$7c,$72,$d7,$fe,$c5,$b6,$00,

$eb,$d1,$cd,$11,$d0,$e0,$d0,$c0,$25,$03,$00,$75,$01,$40,$40,$8b,

$c8,$b8,$00,$00,$b2,$00,$8b,$1e,$36,$7c,$ea,$00,$00,$70,$00,$8d,

$36,$8e,$7d,$eb,$05,$90,$8d,$36,$a2,$7d,$ac,$0a,$c0,$74,$09,$bb,

$07,$00,$b4,$0e,$cd,$10,$eb,$f2,$8d,$1e,$c2,$7d,$3b,$f3,$77,$04,

$8b,$f3,$eb,$e6,$32,$e4,$cd,$16,$8f,$06,$78,$00,$8f,$06,$7a,$00,
 $cd,$19);
 boottext1:
  array[1..136] of char =
        'IBMBIO  COM'+'IBMDOS  COM'+
        'IO    SYS'+'MSDOS  SYS'+
        #10#13'Disk Boot Failure'#00#10#13+
        'Non-system disk or disk error'#00#10#13+
        'Replace and press any key when ready'#10#13;

        (**** constante pentru dischete de 360  kbyte ****)
  form1         = 2;    (* 360 kbyte in unitate de 1,2 Mbyte *)
  media1        = $fd;  (* mediabyte *)
  secperclu1    = 2;    (* sectoare per cluster *)
  radint1       = $070; (* intrari in radacina *)
  secperfat1    = 2;    (* sectoare per FAT *)
  capete1       = 2;    (* numarul de capete (fete) *)
        (**** constante pentru dischete de 720  kbyte ****)
  form2         = 2;    (* 360 kbyte in unitate de 1,2 Mbyte *)
  media2        = $f9;  (* mediabyte *)
  secperclu2    = 2;    (* sectoare per cluster *)
  radint2       = $070; (* intrari in radacina *)
  secperfat2    = 3;    (* sectoare per FAT *)
  capete2       = 2;    (* numarul de capete (fete) *)
        (**** constante pentru dischete de 1,2 Mbyte ****)
  form3         = 3;    (* 1,2 Mbyte in unitate de 1,2 Mbyte *)
  media3        = $f9;  (* mediabyte *)
  secperclu3    = 1;    (* sectoare per cluster *)
  radint3       = $0e0; (* intrari in radacina *)
  secperfat3    = 7;    (* sectoare per FAT *)
  capete3       = 2;    (* numarul de capete (fete) *)
  maxsec        = 17;   (* numarul maxim de sectoare *)

type
  disctab       = array[1..11] of byte;
  disctampon    = array[1..512*maxsec] of byte;
  progmeniu     = array[1..9] of string[20];

var
  tabvechi      : pointer;      (* pointer la vechea tabela a unitatii *)
  eroare        : boolean;      (* indicatia de eroare *)
  verify        : boolean;      (* verificare discheta *)
  coderoare     : byte;         (* codul de eroare *)
  unitate       : byte;         (* numarul unitatii de dischete *)
  selector      : byte;         (* selectia din meniu *)
```

```
  punctmeniu    : byte;          (* punctul de meniu actual *)
 meniu  : progmeniu;     (* linia meniu *)
tabunit : disctab;       (* noua tabela a unitatii *)
tampon : disctampon;     (* tampon de 15 sectoare *)
 pista1,
 pista2,
pista3   : byte;          (* pistele diverselor formate *)
 secperf1,
 secperf2,
secperf3: byte;           (* sectoare per pista *)

(*****************
                    Subrutine
                                  ******************** *)


(* ---------------------------------------------------------------- *)
procedure rama (x1,y1,x2,y2 : byte;  nume : string);
 const
  ss     = $d5;
  ds     = $b8;
  sj     = $d4;
  dj     = $be;
  oz     = $cd;
  vr     = $b3;

 var
  i      :  byte;
  lng    : integer;        (* lungimea sirului cu numele ferestrei *)
   height,
  width : byte;            (* inaltimea si respectiv largimea ferestrei *)
  titlu  : boolean;

 begin
  window(1,1,80,25);       (* tot ecranul *)
   textbackground(black);
  width := x2-x1;          (* largimea ferestrei *)
  height:= y2-y1;          (* inaltimea ferestrei *)
  titlu  := true;          (* fereastra sa primeasca un nume *)
  lng    := length(nume);
  if lng  width then
        titlu := false;    (* renuntam la nume pentru fereastra *)
   textcolor(white);
 if titlu then             (* cind fereastra are titlu *)
     begin
        gotoxy(x1,y1);
        write(ss);         (* coltul din stinga sus *);
        for i := 2 to width do
                write(oz);
        write(ds);
        gotoxy(x2-round(width/2)-round(lng/2)+1,y1);
        write(nume);
     end
  else
    begin
        gotoxy(x1,y1);
        write(ss);
        for i:= 2 to width do
                write(oz);
        write(ds);
    end;
 for i:= 1 to height-1 do
   begin
        gotoxy(x1,y1+i);
        write(vr);
        gotoxy(x2,y1+i);
        write(vr);
   end;
  gotoxy(x1,y2);
 write(sj);                (* coltul din stinga jos al ferestrei *)
 for i:= 2 to width do
        write(oz);
  write(dj);
  window(x1+1,y1+1,x2-2,y2-1);
  textbackground(red);
   clrscr;
 end;

(* ---------------------------------------------------------------- *)

function select(x1,y1,st,ed : byte; var punctmeniu : byte) : byte;

var
 i      : byte;
 lng    :  byte;
 poz    : byte;
 ch     : char;

 begin
        lng := ed-st;
        textbackground(red);
        textcolor(yellow);
        window(x1,y1,x1+20,y1+lng+1);
        for i := st to ed-1 do
                writeln(meniu [i ]);
        write(meniu [ed ]);
         poz := (punctmeniu-st)+ 1;
                        (* pozitia relativa la incep. meniului *)
        gotoxy(1,poz);
        textbackground(white);
        textcolor(black);
        write(meniu [punctmeniu ]);
        repeat
                ch := readkey;
                i := 0;
                case ch of
                        #80:
                                begin   (* sageata in jos *)
                                        gotoxy(1,poz);
                                        textbackground(lightred);
                                        textcolor(yellow);
                                        write(meniu [st+poz-1 ]);
                                        inc(poz);
                                        if poz  lng+ 1 then
                                                poz := 1;
                                        gotoxy(1,poz);
                                        textbackground(white);
                                        textcolor(black);
                                        write(meniu [st+poz-1 ]);
                                end;
                        #72:
                                begin   (* sageata in sus *)
                                        gotoxy(1,poz);
                                        textbackground(lightred);
                                        textcolor(yellow);
                                        write(meniu [st+poz-1 ]);
                                        dec(poz);
                                        if poz  1 then
                                                poz := lng+1;
                                        gotoxy(1,poz);
                                        textbackground(white);
                                        textcolor(black);
                                        write(meniu [st+poz-1 ]);
                                end;
                end;
        until ch=#13;           (* RETURN *)
        punctmeniu := st+poz-1;
        select := punctmeniu;
 end;

(* ---------------------------------------------------------------- *)

procedure clearcur;

 var
  cpu    : registers;
 begin
```

```
  cpu.ah := 1;          (* forma de aparitie a cursorului *)
  cpu.ch := 15;         (* linia de inceput a cursorului *)
  cpu.cl := 0;          (* linia de sfirsit a cursorului *)
  intr($10,cpu);        (* cursorul *)
end;

        (* ------------------------------------------------------------ *)
procedure setcur;

 var
  cpu    : registers;

 begin
  cpu.ah := 1;
  cpu.ch := 11;
  cpu.cl := 12;
   intr($10,cpu);
 end;

(* ------------------------------------------------------------ *)

procedure selectdisc(var unitate  : byte; var meniu: progmeniu);

 begin
  if unitate = 0 then
       begin
              unitate := 1;
              meniu[1] := '  Unitatea: B
       end
  else
       begin
              unitate := 0;
              meniu[1] := '  Unitatea: A  ';
       end;
 end;

(* ------------------------------------------------------------ *)

procedure verifdisc(var verify : boolean; var  meniu : progmeniu);

 begin
  if verify then
       begin
              verify := false;
              meniu[2] := '  Verificare: nu  ';
       end
  else
       begin
              verify := true;
              meniu[2] := '  Verificare: da  ';
       end;
  end;

(* ------------------------------------------------------------ *)

procedure  tabelunitatenou (nrsect, fizic : byte);
              (* creeaza nou tabel ptr. unitate *)

 begin
  tabunit[1] := $df;     (* timpul maximal ptr. schimbarea pistei *)
  tabunit[2] := 2;       (* modul DMA *)
  tabunit[3] := $25;     (* cit mai ramine activ motorul *)
  tabunit[4] := lng;     (* 512 octeti per sector *)
  tabunit[5] := nrsect;  (* cite sectoare / pista *)
  tabunit[6] := $1b;     (* intervalul logic pina la urmatorul sector *)
  tabunit[7] := $ff;     (* lungimea datelor transferate *)
  tabunit[8] := fizic;   (* intervalul fizic pina la urmatorul sector *)
  tabunit[9] := $f6;     (* octetul de umplere *)
  tabunit[10] := 1;      (**)
  tabunit[11] := 8;      (* alti octeti de comanda *)
   getintvec($1e,tabvechi);
  setintvec($1e,@tabunit);
end;

(* ------------------------------------------------------------ *)

procedure tabvechiunit;

 begin
   setintvec($1e,tabvechi);
 end;

(* ------------------------------------------------------------ *)

procedure diskreset;

 var
  contor : byte;
  cpu : registers;

 begin
       contor := retries;
       repeat
              cpu.ah := 0;     (* resetarea unitatilor atasate *)
              cpu.dl := 0;
              intr($13,cpu);
              dec(contor);
       until (contor  1) or ((cpu.flags and 1)  = 0);
 end;

(* ------------------------------------------------------------ *)

procedure formatdisc(form : byte);

 var
  cpu          : registers;
  contor       : byte;

 begin
       contor := retries;
       repeat
              cpu.ah := $17;  (* stabilirea formatului dischetei *)
              cpu.al := form; (* felul formatarii *)
              intr($13,cpu);
              dec(contor);
              if (cpu.flags and 1)  = 1 then
                     diskreset;
       until (contor  1) or ((cpu.flags and 1) = 0);
 end;

(* ------------------------------------------------------------ *)

function formatare(nrpiste, nrsectoare : byte; titlu : string;
       var tampon: disctampon; var errcode : byte) : boolean ;

 type formatrec = record
   trackdisk,
    fatadisk,
   sectordisk,
   octetcontor    : b yte;
  end;

 var
  cpu     : registers;
  contor : byte;
  pista   : byte;
  sector  : byte;
  fata    :byte;
   formproc: byte;
  secformat: word;
  formmax :  word;
  formattab : array[1.. maxsec] of  formatrec;

 begin
       rama(22,19,58,23,titlu);
       formmax := nrpiste * nrsectoare * 2;
```

```
                                (* nr. max. sectoare pe discheta *)
        secformat := 0;         (* nr. sectoarelor formatate *)
        gotoxy(7,2);
        write('% din discheta formatat');
for pista := 0 to nrpiste-1 do
        begin
                for fata := 0 to 1 do   (* doua fete *)
                        begin
                                contor := retries;
                                for sector := 1 to nrsectoare do
                                        begin
        formattab[sector].trackdisk := pista;
        formattab[sector].fatadisk := fata;
        formattab[sector].sectordisk := sector;
        formattab[sector].octetcontor := lng; (* 512 bytes/sector *)
                                        end;
                                textcolor(white);
                                inc(secformat,nrsectoare);
                        formproc := round(secformat/formmax*100);
                                gotoxy(3,2);
                                write(formproc:3);
                                repeat
        cpu.ah := 5;            (* formatare pista *)
        cpu.dl := unitate;      (* numarul unitatii *)
        cpu.dh := fata;
        cpu.ch := pista;
        cpu.al := nrsectoare;
        cpu.es := seg(formattab);
        cpu.bx := ofs(formattab);
        intr($13,cpu);
        if verify then
        begin
                cpu.ah := 4;            (* verificare pista *)
                cpu.dl := unitate;
                cpu.dh := fata;
                cpu.cl := 1;            (* incepind cu sectorul 1 *)
                cpu.al := nrsectoare;   (* verifica toate sectoarele *)
                cpu.es := seg(tampon);
                cpu.bx := ofs(tampon);
                intr($13,cpu);
        end;
        dec(contor);
        if (cpu.flags and 1) = 1 then
                diskreset;
                        until (contor 1) or ((cpu.flags and 1) = 0);
                        if (cpu.flags and 1) = 1 then
                                begin
                                        coderoare := cpu.ah;
                                        formatare := true;
                                exit; (* retur in programul apelant *)
                                end;
                end;
        end;
 formatare := false;
end;

(* ------------------------------------------------------------------ *)

function disc_prep(media,secperclu,radint : byte; nrsectoare : word;
        secperfat,secperpista : byte; var errcod : byte) : boolean;

 var
  contor        : byte;
  numar         : byte;
  i             : integer;
  pointer       : integer;
  cpu           : registers;

 begin
        numar := secperfat*2+round(radint*32/512);
                                        (* sectoare FAT si root *)
        for i := 1 to secperpista * 512 do
                tampon[i] := 0;                 (* sterge tamponul *)
        for i := 1 to 354 do
                tampon[i] := bootsec[i];
    tampon[14] := secperclu;        (* sectoare per cluster *)
    tampon[18] := lo(radint);
    tampon[19] := hi(radint); (* nr. inregistrarilor in radacina *)
    tampon[20] := lo(nrsectoare);
    tampon[21] := hi(nrsectoare); (* nr. total de sectoare *)
    tampon[22] := media;            (* mediabyte-ul *)
    tampon[23] := lo(secperfat);
    tampon[24] := hi(secperfat);
    tampon[25] := secperpista;
        (* sectoare pe o fata intr-o pista *)
    for i := 1 to 136 do
        tampon[i+354] := ord(boottext1[i]);
    tampon[511] := $55;
    tampon[512] := $aa;   (* sfirsitul sectorului de boot *)
    contor := retries;
    repeat
        cpu.ah := 3; (* scrierea bootsectorului pe discheta *)
        cpu.dl := unitate;
        cpu.dh := 0;    (* fata bootsectorului *)
        cpu.ch := 0;    (* pista bootsectorului *)
        cpu.cl := 1;    (* numarul bootsectorului *)
        cpu.al := 1;    (* numarul de sectoare *)
        cpu.es := seg(tampon);
        cpu.bx := ofs(tampon);
        intr($13,cpu);  (* scriere bootsector *)
        dec(contor);
        if (cpu.flags and 1) = 1 then
                diskreset;
    until (contor 1) or ((cpu.flags and 1)= 0);
    if (cpu.flags and 1) = 1 then
        disc_prep := true       (* a aparut o eroare *)
    else
        begin
                for i := 1 to 512 do
                        tampon[i] := 0;
                        (* sterge bootsectorul in tampon *)
                tampon[1] := media;

                tampon[2] := $ff;
                tampon[3] := $ff;
                pointer := secperfat*512;
                tampon[pointer+1] := media;
                tampon[pointer+2] := $ff;
                tampon[pointer+3] := $ff;
                cpu.ah := 3;(* scriere sectoare pe discheta *)
                cpu.dl := unitate;
                cpu.dh := 0;    (* fata dischetei *)
                cpu.ch := 0;    (* numarul pistei *)
                cpu.cl := 2;    (* incepind cu sectorul 2 *)
                cpu.al := secperpista - 1;
                (* numarul sectoarelor ce trebuiesc scrise *)
                cpu.es := seg(tampon);
                cpu.bx := ofs(tampon);
                intr($13,cpu);
                dec(numar,secperpista-1);
                for i := 1 to 512*numar do
                        tampon[i] := 0;         (* sterge tampon *
                cpu.ah := 3;(* scriere sectoare pe discheta *)
                cpu.dl := unitate;
                cpu.dh := 0;    (* fata dischetei *)
                cpu.ch := 1;    (* numarul pistei *)
                cpu.cl := 1;    (* incepind cu sectorul 1 *)
                cpu.al := numar;
                cpu.es := seg(tampon);
                cpu.bx := ofs(tampon);
                intr($13,cpu);
            end;
        if (cpu.flags and 1) = 1 then
            begin
                disc_prep := true;
                diskreset;
            end
```

```
        else
                disc_prep := false;
 end;

(* ---------- *)

procedure numedisc(secperfat,secperpista : byte);
 var
  key   : char;
  nume : string[11];    (* numele dischetei *)
  i     : integer;
  cpu   : registers;
  sector: byte;
  fata  : byte;
  an,
  luna,
  zi,
  ziasaptaminii,
  ora,
  minut,
  secunda,
 sutime : word;
  data,
  time  : word;

 begin
        window(22,18,58,22);
         textbackground(black);
        clrscr;
        rama(16,19,63,25,'Formatare terminata cu succes.');
        gotoxy(4,2);
        write(disksize(unitate+1),' byte disponibili pe discheta');
        gotoxy(12,4);
        write('Continua apasind Enter');
        while key  #13 do
                key := readkey;
        clrscr;
        setcur;
        gotoxy(6,3);
        write('Numele dischetei: [       ]');
        gotoxy(25,3);
        readln(nume);   (* preia numele dischetei *)
        clearcur;
        if length(nume)  0 then
                begin
                        for i := 1 to 512 do
                                tampon[i] := 0;         (* sterge tampon *)
                        for i := 1 to length(nume) do
                                tampon[i] := ord(nume[i]);
                                        (* trece numele in tampon *)
                        tampon[12] := 8;(* atributul pentru nume *)
                        getdate(an,luna,zi,ziasaptaminii);
                                (* preia data din sistem *)
                        gettime(ora,minut,secunda,sutime);
                                (* preia ora din sistem *)
                time := (ora shl 11) or (minut shl 5) or secunda;
                                dec(an,1980);
                                data := (an shl 9) or (luna shl 5) or zi;
                                tampon[23] := lo(time);
                                tampon[24] := hi(time);
                                tampon[25] := lo(data);
                                tampon[26] := hi(data);
                                fata := 0;          (* prima fata a dischetei *)
                                sector := 2*secperfat+2;
                                        (* primul root-sector *)
                                if sector  secperpista then
                                begin
                                        fata := 1;(* fata a doua a dischetei *)
                                        dec(sector,secperpista);
                                        (* sector pe fata a doa a dischetei *)
                                end;
                                cpu.ah := 3;(* scriere sectoare pe discheta *)
                                cpu.dl := unitate;
                                cpu.dh := fata;  (* fata dischetei *)
                                cpu.ch := 0;     (* numarul pistei *)
                                cpu.cl := sector;(* incepind cu sectorul 2 *)
                                cpu.al := 1;
                                (* numarul sectoarelor ce trebuiesc scrise *)
                                cpu.es := seg(tampon);
                                cpu.bx := ofs(tampon);
                                intr($13,cpu);
                        end;
                window(16,19,63,25);
                textbackground(black);
                clrscr;
 end;

(* ---------- *)

procedure mesajeroare(coderoare : byte);

 var
  key    : char;

 begin
        window(22,18,58,22);
        textbackground(black);
        clrscr;
        rama(22,19,58,23,'Eroare!!!');
        gotoxy(8,2);
        if coderoare = 3 then
                begin
                        gotoxy(6,2);
                        write('Discheta e protejata la scriere');
                end
        else
                begin
                        gotoxy(6,2);
                        write('Discheta e inutilizabila');
                        gotoxy(2,3);
                        write('sau nu e de tipul cerut (DD,HD).') ;
                end;
        key := readkey;
        window(22,19,58,23);
        textbackground(black);
        clrscr;
 end;

(* ---------- *)
procedure format360(pista1,secperpista1,fgap : byte; titlu : string);
                (* max. 42 piste *)

 var
  nrsectoare     : word;

 begin
        nrsectoare := capete1*secperpista1*pista1;
                (* nr. total de sectoare *)
        tabelunitatenou(secperpista1,fgap);
        diskreset;
        formatdisc(form1);
  eroare :=  formatare(pista1,secperpista1,titlu,tampon,coderoare);
        if eroare then
                mesajeroare(coderoare)
        else
                begin
        eroare := disc_prep(media1,secperclu1,radint1,nrsectoare,
                secperfat1,secperpista1,coderoare);
                if eroare then
                        mesajeroare (coderoare)
                else
                        begin
                                numedisc(secperfat1,secperpista1);
                        end;
                end;
        tabvechiunit;
```

```
end;

(* ---------------------------------------------------------- *)

procedure format720(pista2,secperpista2,fgap : byte; titlu : string);
                (* max. 84 piste *)

 var
  nrsectoare : word;

 begin
        nrsectoare := capete2*secperpista2*pista2;
                (* nr. total de sectoare *)
        tabelunitatenou(secperpista2,fgap);
        diskreset;
        formatdisc(form2);
        mem [$0040:$0090+unitate] := $54;
                (* comutare pe pas simplu *)
  eroare := formatare(pista2,secperpista2,titlu,tampon,coderoare);
        if eroare then
                mesajeroare(coderoare)
        else
                begin
     eroare := disc_prep(media2,secperclu2,radint2,
                nrsectoare,secperfat2,secperpista2,coderoare);
                        if eroare then
                                mesajeroare (coderoare)
                        else
                                begin
                                numedisc(secperfat2,secperpista2);
                                end;
                end;
        tabvechiunit;
end;

(* ---------------------------------------------------------- *)

procedure format12m(pista3,secperpista3,fgap : byte; titlu : string);
                (* max. 42 piste *)

 var
  nrsectoare : word;

 begin
        nrsectoare := capete3*secperpista3*pista3;
                (* nr. total de sectoare *)
        tabelunitatenou(secperpista3,fgap);
        diskreset;
        formatdisc(form3);
  eroare := formatare(pista3,secperpista3,titlu,tampon,coderoare);
         if eroare then
                mesajeroare(coderoare)
        else
                begin
     eroare := disc_prep(media3,secperclu3,radint3,
                nrsectoare,secperfat3,secperpista3,coderoare);
                 if eroare then
                        mesajeroare (coderoare)
                else
                        begin
                                numedisc(secperfat3,secperpista3);
                         end;
        end;
        tabvechiunit;
end;

(* ---------------------------------------------------------- *)

(*********************
                Programul principal
                                        ********************* *)

begin
        clearcur;
        textbackground(black);
        clrscr;
        rama(20,1,60,17,'Meniulprincipal');
        textcolor(yellow);
        gotoxy(5,2);
        write('MultiFormat pe unitati de 5,25"');
         textcolor(white);
        gotoxy(14,3);
        write('if  Freeware ');
        textcolor(yellow);
        gotoxy(2,4);
         write('Sint disponibile urmatoarele optiuni:');
        gotoxy(4,5);
         write('(sus-jos alege, Enter starteaza)');
         meniu[1] := '  Unitatea        : A  ';
        meniu[2] := '  Verificare: da  ';
        meniu[3] := '   360 kbyte    ';
        meniu[4] := '   420 kbyte    ';
        meniu[5] := '   720 kbyte    ';
        meniu[6] := '   800 kbyte -DD ';
        meniu[7] := '   1.2 Mbyte    ';
        meniu[8] := '   1.4 Mbyte -HD ';
         meniu[9] := '   Terminare    ';
        punctmeniu := 3;
        unitate := 0;
        verify := true;
        checkbreak := false;(* ctrl-break oprit pe durata formatarii *)
        repeat
        selector := select(30,7,1,9,punctmeniu);
        case selector of
        1:
                selectdisc(unitate,meniu);
        2:
                verifdisc(verify,meniu);
        3:
                 format360(40,9,$54,'Formatare discheta 360 kbyte');
        4:
                format360(42,10,$33,'Formatare discheta 420 kbyte');
        5:
                format720(80,9,$54,'Formatare discheta 720 kbyte');
        6:
                format720(80,10,$33,'Formatare discheta 800 kbyte');
        7:
                format12m(80,15,$54,'Formatare discheta 1,2 Mbyte');
         8:
                format12m(82,17,$18,'Formatare discheta 1,4 Mbyte');
        9:
                 ;
        end;
        until selector = 9;
        window(1,1,80,25);
        textbackground(black);
        textcolor(lightmagenta);
        clrscr;
        setcur;
end.
```

Preţuri, preţuri

Rubrica aceasta vă oferă o sinteză a preţurilor pe piaţa germană. În diagramă sînt notate totdeauna cele mai joase preţuri relative la un produs, aşa -numite preţuri de 'stradă' şi de aceea ele pot să se abată mult de la recomandările de preţ ale producătorilor. Preţurile relative la PC-uri se referă la următoarele configuraţii complete, cu placă Hercules şi monitor monocrom:
80286 (12MHz) cu 512 kbyte RAM şi harddisk de 20 Mbyte,
80386SX (16MHz) cu 1 Mbyte RAM şi harddisk de 20 Mbyte precum şi 80386 sau 80486 (25 MHz) cu 2 Mbyte RAM şi harddisk de 40 Mbyte

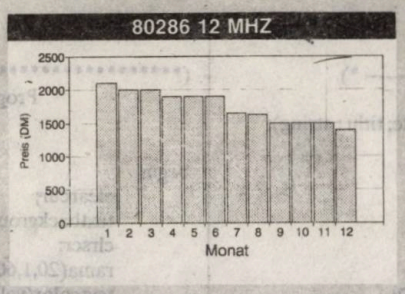
Vă rugăm să aveţi în vedere că produsul cu preţul cel mai avantajos deseori nu este echivalent în ceea ce priveşte compatibilitatea, calitatea şi siguranţa în funcţionare cu produse funcţional echivalente, dar mai scumpe.

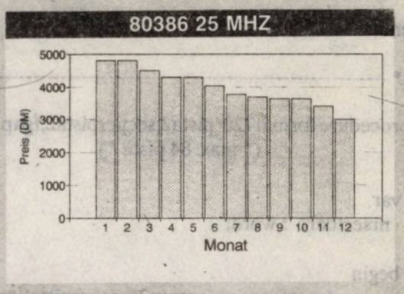
Tendinţe

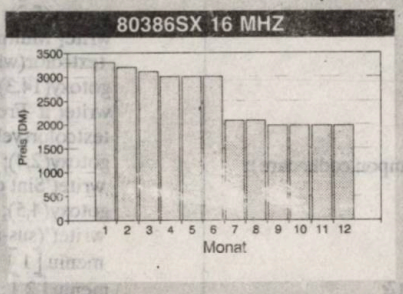
În domeniul imprimantelor Laser compatibile HP este iminentă o cădere a preţurilor. Mulţi distribuitori oferă imprimante Laser sub 2000 DM. Unii producători vor scoate pe piaţă imprimante Laser noi, ieftine în comparaţie cu performanţele relativ ridicate de care vor fi capabile, şi vor să scape de stocurile de modele vechi. Clientul potenţial poate profita acum de scăderile de preţuri, deoarece noile modele probabil nu vor "ieşi" înainte de primăvara '91, cînd preţurile vor fi iarăşi puţin mai ridicate.

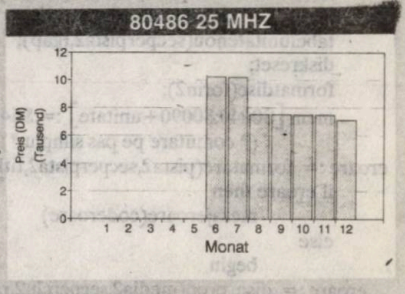
Mic dicţionar german - român:
Farbmonitor = monitor color
Festplatte = harddisk
Nadel-Drucker = imprimantă matriceală

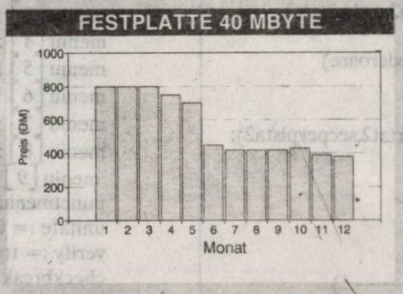
ĂPC + Technik, 1/91Î

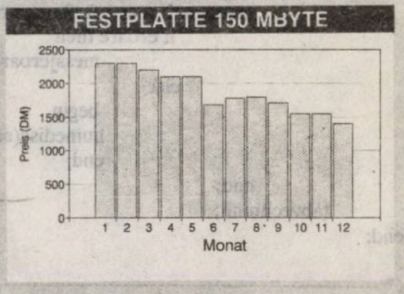


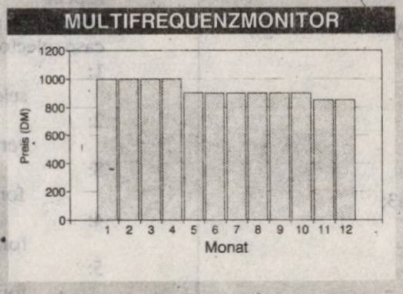


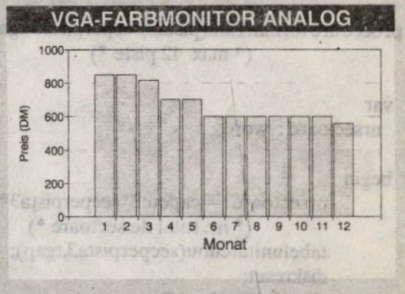


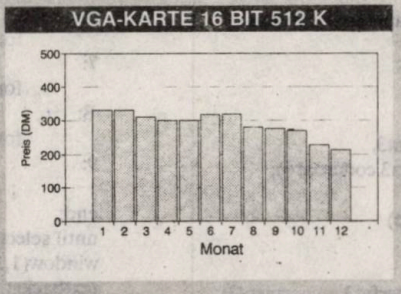


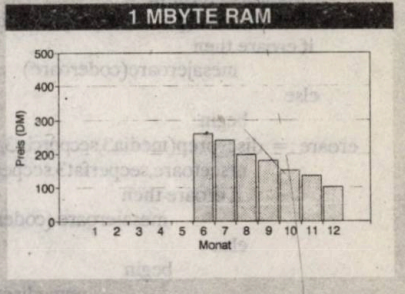


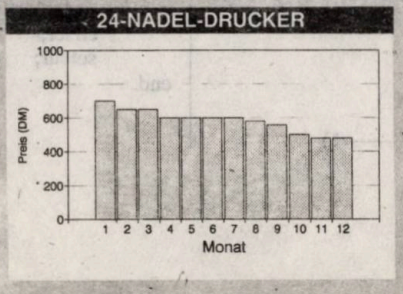


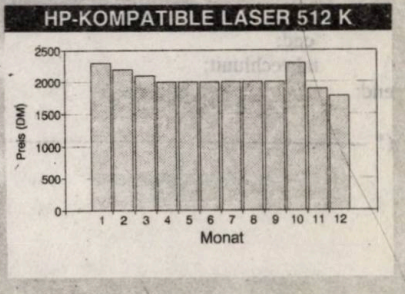












Vreau să cumpăr un AT - de unde şi ce pot să cumpăr?

Aceasta a fost întrebarea cu care am pornit un sumar raid-anchetă telefonic. Cu intenţia de a afla cam care sînt preţurile şi cam cum stă piaţa, am contactat firmele pe care le veţi găsi înşirate mai jos. Desigur, acest raid nu este nici pe departe exhaustiv, şi poate nici foarte reprezentativ: pur şi simplu am testat firmele de a căror existenţă într-un fel sau altul am aflat. Ne va face plăcere să facem publice şi alte oferte, în măsura în care ne vor parveni. Cele de mai jos sînt date relativ la care nu ne asumăm nici un fel de răspundere - le-am notat în dimineaţa zilei de 10 ianuarie 1991, discutînd de la anonim la anonim.

Firma / Telefon	Uc cu	Tact UC (MHz)	Memo UC (Mbyte)	Hard disk (Mbyte)	Ecran	Floppy	Garanţie	Preţ
1)ABTech				**90/10.42.73**				
	286	16	1	40	VGA color	1x5,25"+ 1x3,5"	1 an	cca. 550.000
	386	25/33	2	40	VGA color	"	"	cca. 650.000
Alte amănunte: inclusiv mouse; producţie S.U.A.; calculatoarele vor fi trecute pe la Comisia naţională de standarde								
2)ADMEX				**90/69.20.52**				
	286	12	1	40	VGA color	?	1 an	432.000 lei
Alte amănunte: suplimentar, imprimante 9 ace 80 col. - 72.000 lei								
3)Attigram				**90/33.13.10**				
	286	12	1	40	VGA color	1x3,5" +1x5,25"	1an	390.000 lei +60.000 lei
Alte amănunte: inclusiv maus, MS-DOS 4.01 şi Windows 2.0								
4)BIPO International				**90/~~17.06.26~~** 19.30.89 Laurenţiu Popa				
Alte amănunte: pe moment nu livrează nimic, datorită grevei TAROMS.R.L.								
5)Brodeco				**90/59.05.03**				
	286	12	1	20	EGA mono	1x5,25"		229.000 lei
	"	"	"	"	VGA mono	"		264.000 lei
	"	"	"	"	VGA color	"		304.000 lei
	"	16	"	40	EGA mono	"		320.000 lei
	"	"	"	"	VGA mono	"		361.000 lei
	"	"	"	"	VGA color	"		400.000 lei
	"	25	2	80	VGA color	1x5,25"+ 1x3,5"		692.000 lei
Alte amănunte: inclusiv mouse; 386 are şi 64 kbyte cache; suplimentar se pot achiziţiona: 8087 - 32.000 lei, 80387/16 - 86.000 lei, 80387/25 - 110.000 lei şi imprimante Panasonic 80 col. 192 car./sec. - 57.000 lei, Canon- Epson 132 col. 160 car/sec - 85.000 lei"								
6)Datacom				**951/~~50.878~~** 146.25				
	286		1	40	VGA color	1x5,25"	6 luni	500.000 lei
	"		"	80	"	"	"	580.000 lei
Alte amănunte: inclusiv o imprimantă Epson cu 9 ace								
7)Infocomp				**90/13.26.97**				
	286	16	1	40	VGA color	1x5,25"+ 1x3,5"	1 an	1600 $ sau echivalentul în lei
8)Micro ATCI				**954/17024**				
	286	16	1	40	EGA color	1x5,25"	1 an	510.000 lei
Alte amănunte: inclusiv o imprimantă Epson FX-1000, mouse şi două cutii cu dischete (1 DD, 1 HD)								
9)Microcomputer Service Ltd.				**941/17.414** → 33534 1205 Mădălina				
	286	16	1	40	VGA color	1x5,25"+ 1x3,5"	1 an	650.000 lei
Alte amănunte: inclusiv mouse şi o săptămînă de şcolarizare; în plus, diverse: scannere, streamere etc.								
10)Mons Medius				**994/14173** – Sas. 16894				
	286	8/12	1	40		1x5,25"+ 1x3,5"		210.240 lei
					+ Hercules			29.000 lei
					+VGA color			101.000 lei
Alte amănunte: calculatoarele sînt de producţie RFG, Aquarius Systems International); suplimentar, toată gama de imprimante Star, de la Star SR 10 (48.000 lei) pînă la Star X/2415 (154.000 lei)								
11)Salto S.R.L.				**951/11.929**				
	286	16	1	40	VGA color	1x5,25"+ 1x3,5"	6 luni	710.000 lei
Alte amănunte: inclusiv imprimantă Epson LX800; separat, mouse 12.000 lei								
12)Romtrust				**90/17.08.40**				
"standard"								2400 $ (cca 450.000 lei)

Mică publicitate

Cumpăr HD cu controller IBM-PC compatibil, monitor EGA, 4 MP 8087-2. **Vînd** monitor Orange, set programe, peste 140 (WS, DTP, CAD, Lotus, Harvard Graphics, 3D, Şah, dBase 3+, Mathe, GEM Paint, Write, Ediinf, Fox, Pascal, Prolog, simulator zbor, jocuri, utilitare) 90/717296, 18-22h

"Sava Soft" Buzău vă oferă:
-salarii .... CP/M, MS-DOS
-balanţa de verif..... CP/M, MS-DOS
-mijloace fixe ... . CP/M, MS-DOS
-facturi-livrări . ..... CP/M, MS-DOS
-stocuri (în preg) .... MS-DOS
ing. Sava Stan, tel. 974/12760

Doresc colaborare sau scurte consultaţii pt. **Commodore 64** în special probleme legate de cod maşină Florin 90/803384

A apărut revista lunară de High Fidelity, electronică audio şi software pentru generare de sound şi compoziţie muzicală. Abonamente la orice oficiu poştal. Adresa redacţiei: Trever Emil, str. Topliţa nr.7, 1900 Timişoara, jud. Timiş, tel. 961/48631. Citiţi şi veţi fi mulţumiţi: color, 36 pagini.

SINTES 6

Universitatea din Craiova, Facultatea de automatică şi calculatoare, organizează ediţia a şasea a simpozionului naţional de teoria sistemelor, robotică, calculatoare şi informatică de proces. Simpozionul se va desfăşura în perioada 14 - 15 (!) martie 1991; Tematica destul de largă cuprinde printre altele:
-conducerea adaptivă a roboţilor
-echipamente pentru testare metrologică
-conducerea optimală, cu calculatoare numerice, a proceselor din energetică şi industria chimică
-procesoare dedicate pe aplicaţii
sisteme numerice cu prelucrare paralelă, transputere
În timpul simpozionului, firmele Tektronix, Hewlett-Packard precum şi firmele sponsorizatoare vor expune unele din produsele lor.
Alte informaţii: Cernian Oleg, Universitatea din Craiova, Facultatea de Automatică şi Calculatoare, Catedra de Calculatoare, str. Lăpuşului nr. 5, 1100 Craiova, tel. 941/45724, int. 168,169

Din cuprinsul numărului viitor:
- Programe de calcul tabelar (spreadsheet-uri)
- Microprocesorul RISC intel 860
- State-of-the-art: cele mai novatoare 5 concepte hard şi soft ale anului 1990: digital video interface, TFT-LCD, MS-Windows 3.0, tehnologia flash, programarea orientată obiect **Şi încă multe altele!...**

Poşta redacţiei

Începem prin a ne scuza de două ori: întîi, în faţa colaboratorului nostru, ing. Doboş Ştefan, care a făcut toate desenele din revistele noastre şi pe care l-am omis sistematic din caseta redacţiei; apoi, în faţa cititorilor, deoarece din considerente tehnice articolul relativ la microprocesorul RISC intel860 va apare abia în numărul viitor şi nu în acest număr, cum a fost anunţat.

Mulţumim tuturor celor care ne-au scris de anul nou - şi tuturor celor care ne scriu în general. Nu numai că ne ajută, dar ne sînt chiar indispensabile reacţiile la ceea ce punem în revistă: doar o facem să fie citită şi, mai ales, să fie utilă.

Pentru a simplifica procedura de expediere, destul de laborioasă la multe expedieri cu ramburs, dorim să simplificăm procedura şi să trecem la expedierea de recomandate cu confirmare de primire. Pentru aceasta, rugăm abonaţii noştri să ne trimită contravaloarea revistelor în avans, pentru numărul de luni care îi interesează. Plecăm de la premisa că vom reuşi să păstrăm preţul revistei la 39 lei pe număr. Cei care doresc să se aboneze în acest fel sînt rugaţi să trimită mandate poştale cu confirmare de primire pe adresa redacţiei, pentru Ingrid Maier. (Confirmările ar trebui păstrate, ele fiind "chitanţa" de abonament). O menţiune suplimentară pe mandat, relativă la faptul că cel care-l trimite e un abonat mai vechi care renunţă la forma veche de abonament şi trece pe forma nouă sau este un abonat nou poate numai să ne ajute.

Va fi mai simplu în acest fel şi la expediere, dar şi la primire, deoarece recomandatele (mai) vin acasă, fără să necesite (totdeauna) un drum pînă la poştă. Îi mai rugăm pe cei care doresc să facă abonamente să le facă pe 4 sau pe 10 luni, abonamentul fiind astfel valabil pînă la jumătatea anului sau respectiv pe tot anul 1991; primul număr care va veni pe abonament va fi numărul 3/1991; pînă atunci, nu se schimbă nimic. Mulţumim celor care ne vor ajuta să mai scăpăm un pic de formularistică, plătind anticipat, dar şi tuturor celor care ne pot ajuta cu sugestii sau propuneri valabile relativ la simplificarea difuzării. Un caz fericit, dar singular, îl avem la Baia Mare, unde mai mulţi cititori s-au abonat la un difuzor local (dl. Cudalbu Octavian, tel. 994/36433); şi pentru noi, acest lucru reprezintă o uşurare semnificativă, deoarece nu mai trimitem 50 de plicuri ci un singur colet, iar în evidenţele noastre figurează un singur abonat.

(ing. Iosif Fettich)

6. Urmatoarele articole nu mi-au plăcut:

...

...

...

datorită |_| subiectului |_| formei de prezentare

7. Calificativul acordat acestui număr: .....

8. Propuneri de îmbunătăţire:

...

...

.9. Mi-au displăcut mai multe exprimări/termeni folosiţi în revistă. Propun ca în locul lor să se folosească un alt echivalent.

Termen	Echivalent
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

Data..... Semnătura..... if 1 / 91

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI

C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI

C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi lunar |_| |_| |_| exemplare ale revistei **if,** începînd cu numărul |__|, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

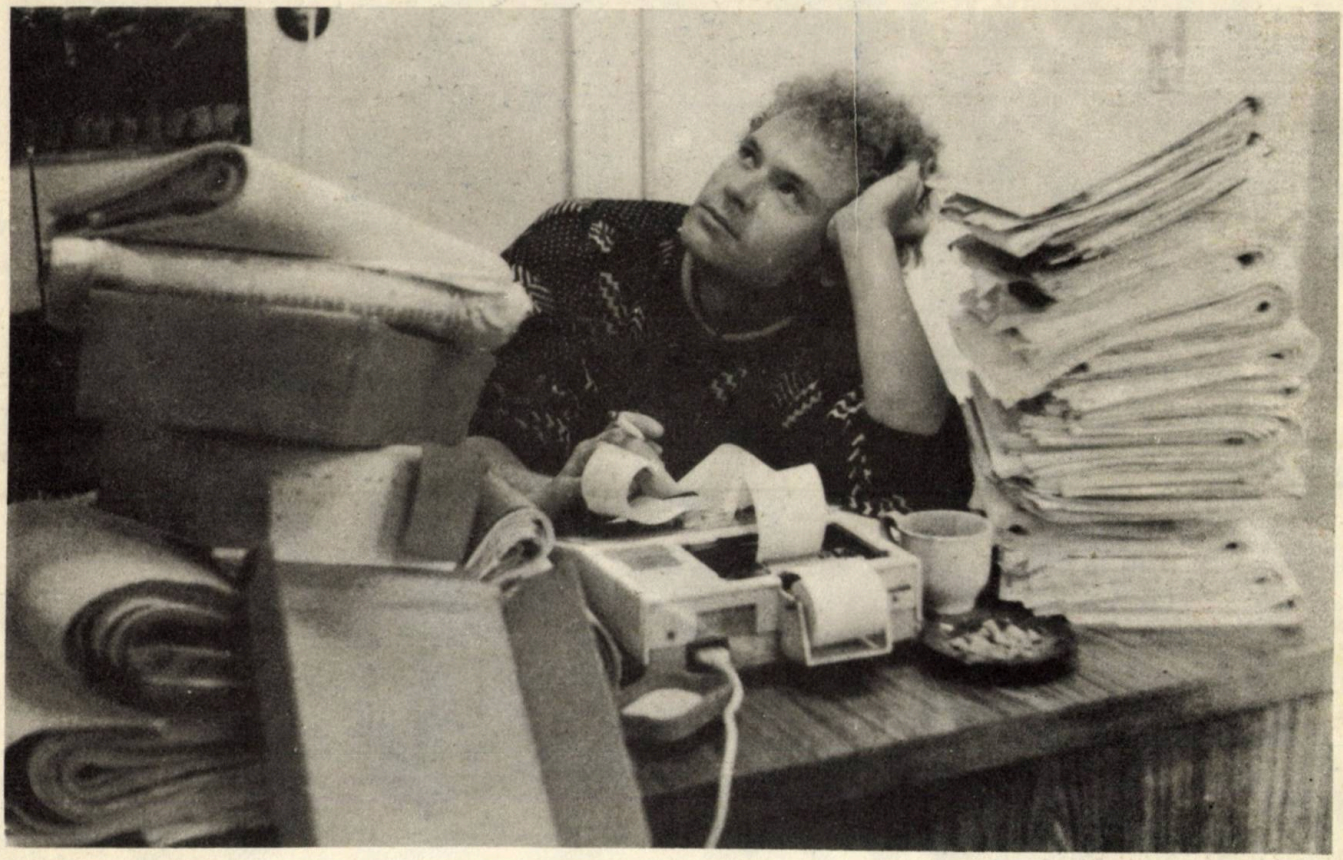
Preţul unui exemplar este 39 lei. Plata se face prin ramburs. Anularea comenzii se face prin scrisoare recomandată sau cu confirmare de primire şi intră în vigoare din luna următoare.

if 1 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_



Don't worry, be happy !

Micro ATCI Tirgu Mures

va ofera o alternativa !

Daca doriti sisteme IBM PC la cheie
(consulting, instalare, asistenta tehnica)
contactati-ne la adresa :

Micro ATCI Tirgu Mures

C.P. 64

RO - 4300 Tirgu Mures

Tel. 954 / 17024

IEEE-488 interface systems

(IEEE-488-2/IEC-625-2, HP-IB, GPIB)

Calculator: IBM PC/XT/AT/IC/386/486, IBM PS/2 etc.
Sistem de operare: MS-DOS, OS/2 386™, UNIX, XENIX etc.
Compatibil cu HT-Basic, ASYST/ASYSTANT GPIB, Lotus Measure, GURU, LabWindow, DaDisp 488etc.
Rate de transfer: 100-200kbyte/sec (DOS),
35kbyte/sec (OS/2,UNIX),
330-500kbyte/sec (8-bit DMA),
1Mbyte/sec (16bitDMA)

Garanție: 5 ani

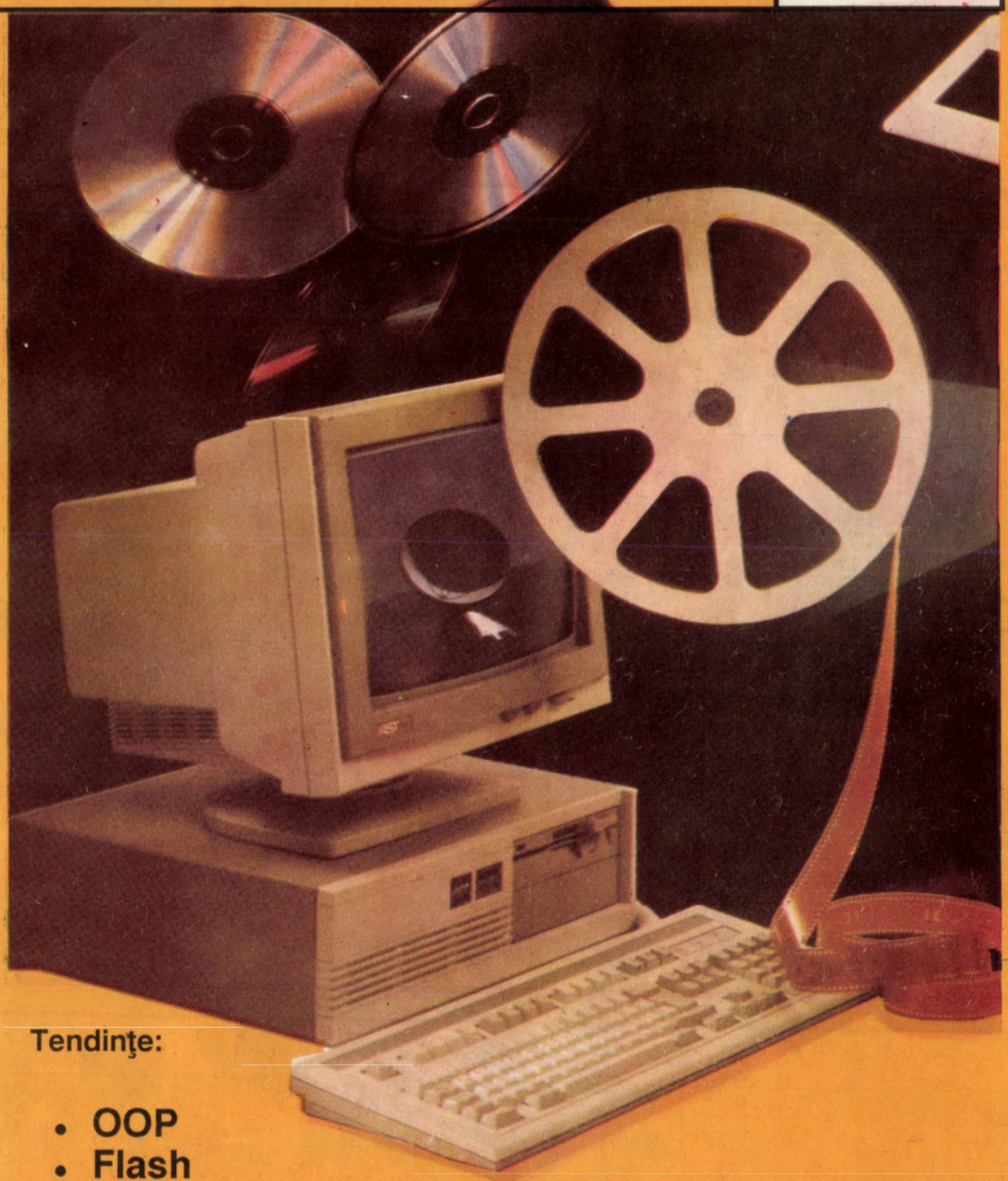
ines - INnovative Elektronik Systeme GmbH
Neuenhöfer Allee 45, 5000 Köln 41, Germania

if

CALCULATOARE PERSONALE

2 / 91
(5)
FEBRUARIE
39 Lei

Revistă lunară editată de Micro ATCI Tîrgu Mureş



Tendinţe:

- OOP
- Flash
- TFT-LCD
- Multimedia
- Windows 3.0

Viruşi

Calcul tabelar

if if if if

GEOS
AMIGA
REFLECTIONS
AutoCAD
EXCEL
WORKS
SIGNUM!
GEOCHART

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

La realizarea acestui număr au colaborat:

ing. Attila Darvas,
ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,
mat. Eugen Rotaru
Colaboratori externi:
fiz. Vera Radovici
stud. Ioan T. Socaciu
Desene: ing. Ştefan Doboş

Tiparul: tipografia Tîrgu Mureş
Tipografi: Halaţiu Mihai,
Cormoş Mircea

Revista apare lunar.

Preţ: 39 lei

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/17024, fax 954/35208.

Manuscrise originale sau listing-uri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu directorul. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate la următoarele tarife:
1 cm^2 (alb-negru) 30 lei,
1 cm^2 (alb-negru şi o culoare suplimentară) 40 lei,
1 cm^2 color 60 lei.
Doritorii sînt rugaţi să ia legătura cu redacţia.

Femartie

Sperăm ca acest număr de februarie să apară măcar în martie şi să poată constitui un plăcut cadou pentru gingaşele noastre cititoare. În ciuda eforturilor depuse pînă acum, de a scoate revista pe piaţă cu o periodicitate lunară, întîrzierea cu care apar numerele creşte. Mai sîntem încă suficient de naivi ca să sperăm că vom recupera întîrzierea şi vom intra în normal, şi nu avem de gînd să depunem armele. Şi dacă pînă acum am bombardat tipografia cu materiale, în curînd vom începe ofensiva terestră. Deocamdată revista apare, cînd iese de la tipografie.

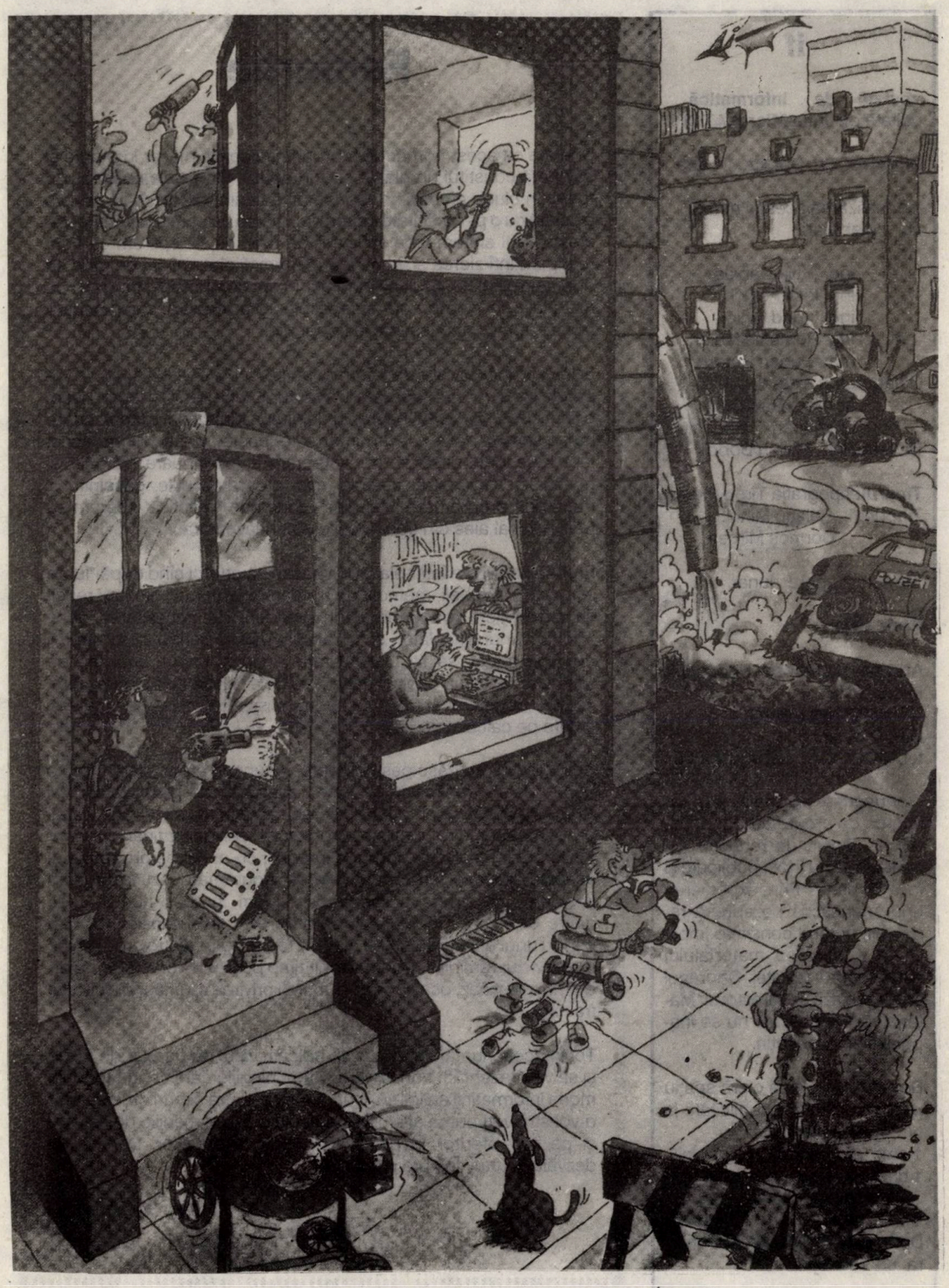
Editarea unei reviste de informatică pe coordonatele spaţio-temporale pe care ne aflăm este o aventură. Am pornit la drum, avem o ţintă clară, dar avem nevoie şi de ajutor. Mulţumim cititorilor care ne-au scris şi dacă acest număr este, poate, mai echilibrat acest fapt li se datorează şi lor. Aşteptăm în continuare sfaturile şi materialele d-voastră; articole, mici programe utile, găselniţe şi trucuri de programare. Mai ales rubrica de "practică" sperăm să trăiască mai ales din contribuţiile dumneavoastră.

Probabil că vi s-a întîmplat şi d-voastră ca atunci cînd după "lupte seculare", hărţuieli, memorii de fundamentare, etc., etc., aţi reuşit să convingeţi conducerea întreprinderii de necesitatea achiziţionării unui calculator, şi aţi răsuflat uşurat pentru că în sfîrşit îl aveţi pe masă, să fiţi pus cu faţa la zid şi supus unui foc de reproşuri: "că s-au cheltuit atîţia bani pe el" şi "că nu l-aţi cumpărat să stea degeaba" şi "că imediat să-mi scoateţi situaţiile cutare şi cutare pe calculator".

Aplicaţiile nu se nasc peste noapte, iar uneori efortul care trebuie depus pentru realizarea lor este neaşteptat de mare; totuşi, cu ajutorul programelor de calcul tabelar (spreadsheet-uri) puteţi preîntîmpina multe conflicte. Problemele serviciilor financiar-contabile pot fi rezolvate, aproape integral, cu astfel de programe şi aceasta aproape fără a fi nevoie să programaţi, rezultatele fiind imediate. Din aceste motive vă prezentăm, în acest număr, şase dintre cele mai răspîndite programe de calcul tabelar. Optînd pentru unul din ele vă puteţi rezolva rapid multe dintre problemele de serviciu, rămînîndu-vă mai mult timp pentru d-voastră. Nu pierdeţi, în acest caz, ocazia să vă bucuraţi privirea cu imaginea cîtorva fractali.

Tot în acest număr încercăm să ridicăm vălul de pe ziua de mîine şi să vă prezentăm conceptele şi tehnologiile care vor da tonul în moda informaticii anilor 90. Cum puteţi asigura securitatea datelor d-voastră veţi putea afla citind articolul despre mediile de salvare şi pe cel referitor la protecţia împotriva viruşilor. Dar să nu dezvăluim totul, lectură plăcută !

ing. Romulus Maier



"Nu ţi se pare şi ţie că ventilatorul e cam zgomotos?"

Cuprins

Editorial . . . . . pag. 1

Ştiri

- Discurile de 3,5" în pericol! . . pag. 4
- Splendoare în pixeli . . . . . pag. 4
- 486 la 50 MHz . . . . . pag. 4
- Muzică cibernetică . . . . . pag. 4
- Tablă minune . . . . . pag. 4
- Un nou Touchscreen . . . . . pag. 4
- Tapol preia PC Data . . . . . pag. 5
- La bunăvoinţa publicului . . . . . pag. 7
- Staţie de lucru . . . . . pag. 9
- Calul troian ? . . . . . pag. 9

Software

Atlas soft
Înconjurul lumii în 60 de minute . . . . pag. 11

Aşa geografia este o plăcere : Cu "PC Globe" şi cu calculatorul într-o expediţie în jurul lumii.

Remote-Control Software . . . . . pag. 12

- Carbon Copy Plus . . . . . pag. 13
- Co/Session . . . . . pag. 13

Calcul tabelar . . . . . pag. 14

- Lotus 1-2-3 2.2 . . . . . pag. 15
- Excel 2.1 . . . . . pag. 16
- Lotus 1-2-3 3.1 . . . . . pag. 17
- Supercalc 5 . . . . . pag. 18
- Quattro Pro . . . . . pag. 19
- Starplaner 2.0 . . . . . pag. 19

Fundamente

Haos şi fractali . . . . . pag. 21

Despre atractori "ciudaţi", cu dimensiuni fracţionare şi despre autoasemănare. Istoria haosului, ordinii şi a fractalilor.

Vă prezentăm

Un Cray pe 1,5 centimetri pătraţi - procesorul Intel i860 . . . . . pag. 26

Tendinţe

State of the art . . . . . pag. 29

Sub titlul "State of the art" vă prezentăm 5 concepte hard şi soft care în opinia noastră, vor juca în viitor un rol decisiv în lumea calculatoarelor

- Multimedia Show . . . . . pag. 30
- Ecranul viitorului: TFT - LCD . . . . . pag. 32
- Ferestre în lumea datelor . . . . . pag. 34
- Tehnologia flash
  Mediu de memorare de viitor . . . . pag. 36
- Programare orientată obiect
  Transparenţă prin "clase" . . . . . pag. 38

Service

Fascinaţia viruşilor . . . . . pag. 40

- Lista principalilor viruşi cunoscuţi . . pag. 44

Hardware

O salvare corectă . . . . . pag. 46

Salvările (backup-urile) sînt importante, ele sînt folositoare însă numai atunci cînd şi strategia de salvare este corespunzătoare. În această privinţă vă putem da cîteva sugestii.

Practică

- O ghilotină pentru fişiere . . . . . pag. 49
- Scrierea unui număr în litere . . . . pag. 52

Ultima pagină

- Izvoditoriul cititorului, sănătate . . . pag. 54

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Caricatura . . . . . pag. 2
- Poşta redacţiei . . . . . pag. 10
- Mica publicitate . . . . . pag. 54

486 la 50 MHz

Primul calculator cu un procesor Intel 80486 şi care lucrează la 50 MHz a fost prezentat de firma americană Transcomputer. S-a pornit de la o placă de bază realizată cu 80386 la 25 MHz. Transcomputer a realizat o plachetă care se implantează în soclul procesorului 80386 şi are în partea superioară un soclu pentru 486. Răcirea necesară la această frecvenţă de deas ridicată este asigurată de un corp de răcire special, care împiedică supraîncălzirea procesorului.

Tablă minune

Gata cu copiatul! Panasonic a elaborat o tablă care, la apăsarea unei taste, se copiază singură. Ceea ce este pe tablă se reproduce pe pagini A4 şi poate fi împărţit la toţi participanţii la curs, lecţie sau seminar. Avantajul: ascultătorii se pot concentra asupra prelegerii, nu mai trebuie să facă efortul de nota ce se scrie pe tablă. "Panaboard" se cheamă tabla minune şi costă circa 2700 mărci. Informaţii suplimentare: Panasonic Deutschland, Winsbergring 15, 2000 Hamburg 54.

Muzică cibernetică

O aplicaţie ieşită din comun a unei reţele au demonstrat recent Goupil Deutschland împreună cu Bose, firmă producătoare de difuzoare şi casa de soft Ploenzke Informatik. În cadrul zilelor muzicii religioase europene din Schwäbisch Gmünd, H. Bledsnajder a concertat pe o orgă cuplată cu 4 orchestre "virtuale". Aproape 33.000 de difuzoare, patru sisteme de trecere a frecvenţelor joase, sintetizatoare precum şi 4 laptop-uri şi un 386 au fost conectate între ele pentru a forma un sistem de calcul paralel. Softul "cibernetic" traducea semnalele preluate de la orgă, pentru a le transmite la calculatorul central, care la rîndul lui deservea cele 4 laptop-uri.

Discurile de 3,5" în pericol!

Un nou studiu despre discurile de 3,5" (high-density) a descoperit că multe din discurile de pe piaţă erau bănuite de utilizatori ca fiind proaste. Studiul condus de corporaţia Memory Control Technology (MemCon) a întocmit un clasament cu primele 8 din 21 de mărci de discuri la fel de "bune". 5 mărci, inclusiv cele vîndute de corporaţia IBM, au fost cotate de nivel mediu sau mai slab.

MemCon a testat 100 de discuri din fiecare marcă folosind un echipament special de testat discuri, urmărind 11 criterii de cotare după o scală de la 2 (foarte slab) la 10 (excelent). Testul a fost bazat pe metode recomandate de American National Standard Institute. Raportul arată că, dintre toate mărcile, numai Sony a trecut cu succes toate cele 100 de discuri prin două teste: "missing bits" şi "extrabits test". Topul alcătuit de MemCon arată cum urmează: 1. Sony 2.Fuji 3.3M 4.Kodak 5.Memorex 6.BASF 7.Maxell 8.Polaroid
Surpriză? (după Electronic World News, 24/90)

Un nou Touchscreen

În toamna anului trecut, la COMDEX, giganticul tîrg de calculatoare de la Las Vegas, IBM a prezentat o nouă realizare de senzaţie: un ecran care sesizează atingerile şi lucrează în 3 dimensiuni. Adică: nu sesizează numai unde se apasă (coordonate X şi Y) ci şi cît de tare (coordonata Z). Cu ajutorul unui astfel de ecran, programele pot interacţiona cu utilizatorul de o manieră mult diferită faţă de metodele cunoscute. De exemplu, parcurgerea unei baze de date poate fi grăbită apăsînd un pic mai tare. Se pot imagina şi programe de desenare în care un "creion" electronic - ca în viaţă - desenează diferit funcţie de apăsare. Noua tehnică a fost brevetată, căci, în plus, noul ecran nu se uzează. "Touchscreen"-urile mai vechi au o plasă metalică fină pe deasupra ecranului, pentru a măsura atingerea. Conductorii extrem de subţiri pot fi distruşi după un număr de apăsări. IBM utilizează tipuri noi de senzori de mişcare, montaţi în cele 4 colţuri ale ecranului. Cum anume lucrează, a rămas încă un secret al firmei. Secret este încă şi preţul; surse inoficiale îl estimează la circa 1400$.

Splendoare în pixeli

La sfîrşitul lui ianuarie 1991 a fost decernat pentru a zecea oară premiul "INA Pixel". În cadrul "tîrgului" european de artă pe calculator, "Imagina", (30 ianuarie - 1 februarie, Monte Carlo), fundaţia franceză INA a acordat 10 premii pentru următoarele domenii: ficţiune, cercetare, simulare, animaţie, efecte speciale, publicitate, televiziune, microcalculatoare, şcoli şi universităţi. În afară de aceasta, studenţii pot cîştiga o bursă pentru producerea de imagini artificiale tridimensionale dacă participă la concursul fundaţiei Ricard. Informaţii suplimentare: INA, Pierre Hénon, Délégation Stratégie et Developpement, Pice 3224, 4 Avenue de l'Europe, 94366 Bry-sur-Marne, France.

Tapol preia PC Data

Prin vînzarea de software de marcă la preţuri absolute de dumping, întreprinderea PC Data a ajuns pe prima pagină a ziarelor şi în atenţia procuraturii. Dacă pînă acum PC Data se ocupa de copii pirat servindu-se de o adresă de acoperire din Arabia Saudită, de scurt timp clienţii sînt deserviţi din Varşovia.

Cu toate că PC Data se ocupă de cîtăva vreme cu vînzarea ieftină de copii pirat de programe standard încă nu s-a dat de urma făptaşilor. Prin crearea de diferite staţii în mersul afacerii urmele au fost şterse pînă în prezent aproape cu succes. Comenzile urmau procedura următoare: un client interesat trebuia să-şi trimită comanda în Arabia Saudită. De acolo ea era expediată din nou în Germania unde era prelucrată.

Pachetul cu dischetele care conţineau programul comandat erau recepţionate de client ca fiind expediate din Germania, fostul RDG, sau chiar Olanda, de fiecare dată fără expeditor. Atunci cînd clientul plătea cu un cec cu suma integrală, primea dischetele care puteau fi imediat utilizate. Dacă plata se făcea ulterior (după nume), clientului îi erau expediate dischetele cifrate, discheta cu cheia primind-o numai după achitarea integrală a sumei.

Totuşi clienţii n-au primit numai software de marcă ieftin, şi ilegal, ci au mai primit şi viruşi şi probleme, deoarece niciunul dintre clienţi nu putea fi sigur că aceste copii ilegale erau nevirusate. Nu de puţine ori se întîmpla ca programul, după instalare, să nu meargă. "Dar piraţii de soft nu se ocupă cu astfel de probleme şi atunci utilizatorii cer ajutor de la producători" ştie din proprie experienţă Reinhard Gründer, director la Digital Research.

Clienţii înşelaţi şi producătorii păgubiţi au cerut intervenţia procuraturii. "În acest caz investigăm deja de relativ mult timp", relatează Dieter Wolf însărcinatul cu investigaţiile din partea procuraturii din Köln. Pînă acum n-am putut descoperi oamenii care se ascund în spatele firmei PC Data. "În cursul investigaţiilor noastre am putut afla un nume şi o adresă din Arabia Saudită, dar acolo n-am putu face cercetări deoarece drepturile de autor sînt altfel legiferate, iar mîinile autorităţilor erau legate", mărturiseşte în continuare Wolf.

În cele din urmă s-a putut stabili cu certitudine că adresa din Arabia Saudită nu era decît o adresă de acoperire pentru cei de la PC Data. "Comenzile erau reexpediate în Germania şi de aici erau onorate", explică Wolf, dar adresa exactă n-a putut fi stabilită pînă în prezent. "Am făcut mai multe presupuneri şi ştim că multe pachete au fost expediate din regiunea Koblenz, dar încă nu putem afirma nimic concret. Cercetările continuă în această regiune", îşi caută Wolf o acoperire.

Situaţia s-a modificat cu ceva, deoarece nu mai este PC Data cea care oferă soft de marcă la preţuri de dumping, ci Tapol. Această întreprindere îşi are sediul în Polonia şi a expediat foştilor clienţi sau interesaţi ai lui PC Data o nouă listă de preţuri. În această listă Word 5.0 este oferit pentru 185 mărci, Pagemaker pentru 240 mărci, iar Autoshade pentru 610 mărci. Pe cale legală Word 5.0 se poate cumpăra pentru 1700 mărci, Pagemaker pentru 2800 mărci, iar Autoshade pentru 2400 mărci.

Într-o înştiinţare pe care Tapol a expediat-o în momentul "preluării afacerilor" se anunţa că toate obligaţiile de livrare rămase de la PC Data vor fi satisfăcute, motivul preluării fiind vag explicat printr-o interdicţie de vînzare în Europa impusă firmei PC Data. În acelaşi timp întreprinderea asigura că utilizarea soft-ului obţinut de la diferite surse este legală şi că aşa cum prevăd normele CSS (Continental Surplus Supply), programele au fost testate "cu toate mijloacele aflate la dispoziţie" împotriva viruşilor şi că nu sînt infestate.

Intreprinderea mai face încă un pas şi oferă clienţilor săi o posibilitate de cîştig pentru transmiterea altor comenzi. "Cîştigaţi alături de noi", se spune la sfîrşitul listei de preţuri. Clienţii care aduc beneficiari noi vor primi 3% din vînzarea brută. Tapol pretinde discreţie absolută, chiar dacă clientul va primi în mod regulat o listă a beneficiarilor aduşi de el.

Modul în care Tapol respectiv CSS îşi serveşte clienţii seamănă extrem de mult cu cel al lui PC Data. După părerea lui Johannes Kröger, avocat şi conducător de afaceri al Uniunii pentru Dezvoltarea Industriei Germane de Software (VSI), este vorba de aceeaşi întreprindere. "PC Data trebuia să dispară din Arabia Saudită, după ce acolo legile s-au schimbat, iar acum operează din Polonia". Şi aici comenzile sînt preluate de către un om de paie şi transmise în Germania, iar pachetele cu marfa sînt expediate fără adresa expeditorului. Şi întreprinderea Tapol lucrează cu programe cu parolă şi dischete protejate prin cheie."Tapol este doar o pseudo-întreprindere. În Polonia doar cîteva întreprinderi au aprobare pentru comerţ exterior. După cum se vede, Tapol are o asemenea aprobare, pe care PC Data a folosit-o pentru afacerile ei", afirmă Kröger.

În orice caz, se pare că Tapol are deja necazuri."De circa trei săptămîni am încercat trei comenzi de test, dar nici pînă astăzi n-au reacţionat", ne informează con-

ducătorul de afaceri al VSI Kröger. "N-am primit nici o înştiinţare şi nici de la bancă n-au fost extraşi bani".

Dar chiar dacă în prezent întreprinderea poloneză se arată puţin activă, VSI continuă investigaţiile. "Împreună cu Business Software Alliance BSA şi cu ambasada americană din Varşovia, VSI încearcă în prezent să determine guvernul polonez să ia măsuri împotriva întreprinderii Tapol. În acelaşi timp verificăm şi situaţia drepturilor de autor în Polonia. În acest scop tocmai se traduce legea. Problema este în orice caz de durată".

Kröger nu vede probleme doar în domeniul juridic; şi situaţia poliţienească este dificilă. "Din punct de vedere al poliţiei, situaţia este foarte greu de apreciat. Poliţiştii nu sînt instruiţi în acest sens şi le lipseşte dotarea necesară. Poliţia este practic depăşită. Dintr-o dată ea este confruntată cu o criminalitate, pe care n-a cunoscut-o pînă acum şi cu care nu ştie să lupte". Dar şi Polonia poate deveni nesigură pentru Tapol, CSS sau PC Data. "Pînă în prezent Polonia n-a avut o lege a drepturilor de autor", ne explică procurorul Wolf. "Dar situaţia legală se va schimba în curînd, în scurt timp va exista o asemenea lege".

Activităţile lui PC Data respectiv Tapol sînt cunoscute nu numai de procuratură şi de VSI, ci şi cei mai mulţi producători de software cunosc întreprinderea şi maşinaţiile ei. "PC Data vinde copii, ceea ce este ilegal, şi nu este bine", ne lămureşte Dirk Seebald, conducător de afaceri la Wordperfect.

Producătorul de soft nu vrea să accepte maşinaţiile lui PC Data sau Tapol. "Sîntem membri ai BSA şi ai VSI şi sprijinim activităţile acestor organizaţii. E de la sine înţeles că sîntem interesaţi să se pună capăt cît mai repede acestei activităţi". Paguba care a suferit-o Wordperfect prin PC Data nu se poate estima aşa de uşor. "În mod cert paguba datorită afacerilor lui PC Data nu se ridică la milioane, dar global ea este totuşi mare". El crede că PC Data nu mai are succesul de dinainte: "munca noastră de lămurire a dat roade la utilizatorii de PC-uri, şi fiecare ştie astăzi, ce se poate întîmpla şi ce amenzi poate primi dacă face asemenea afaceri şi cumpără soft furat". Seebald crede că clienţii principali a lui PC Data se găsesc mai ales printre cei care lucrează acasă. "Cumpărătorii sînt recrutaţi din domenjul utilizatorilor de PC, care oricum colecţionează acasă copii pirat". După părerea sa întreprinderile nu fac parte din cercul clientelei lui Tapol sau PC Data. "Nici pe studenţi nu mi-i pot imagina ca şi clienţi ai lor, deoarece condiţiile pe care le oferim la facultăţi sînt atît de bune, încît nu merită să cumpere copii pirat. Programele pentru studenţi, cu licenţă regulată, costă doar cîteva sute de mărci. Produsele oferite de PC Data sînt doar cu puţin sub preţurile noastre şi în plus nu sînt legale".

Şi Aldus din Hamburg cunoaşte activităţile piraţilor de soft. Ca membru al VSI el sprijină iniţiativele organizaţiei în mod concret. "Nu prea are sens să întreprinzi acţiuni mari de unul singur", este de părere Manuela Steffen, partener de dialog în probleme de presă de la Aldus. "Acţiunile singulare necesită multe resurse, care de multe ori nu pot fi obţinute". În schimb dacă la Aldus se primesc informaţii că undeva circulă copii pirat, atunci responsabilii devin activi. "Dar foarte rar piraţii de soft ne indică ce au de gînd cu copiile", afirmă ironic Manuela Steffen.

Nici la Digital Research acţiunile singulare nu sînt văzute cu ochi buni, pentru că trebuie investit prea mult şi se pot contabiliza prea puţine rezultate. Şi Digital Research a activat împotriva lui PC Data. "Am efectuat deja o cumpărare de test, am obţinut chiar şi semnătura celui care a obţinut contravaloarea ceculuj, dar în final n-am reuşit să aflăm nimic concret" ne spune purtătorul de cuvînt al casei. Întreprinderea a oprit eforturile proprii în momentul în care VSI a luat iniţiativa în această problemă.

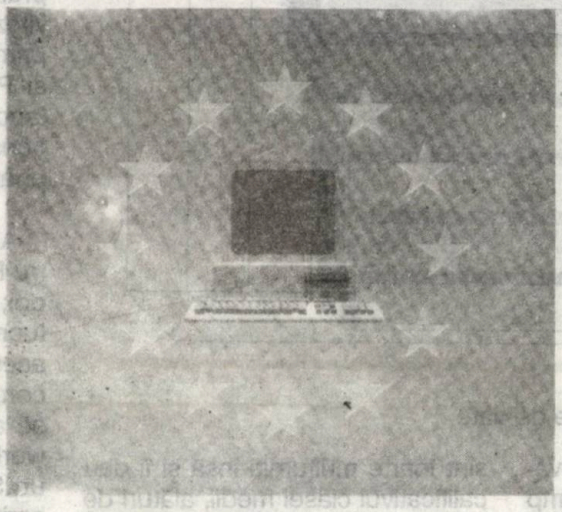
Membrii VSI Lotus din München şi Ashton Tate din Frankfurt se bazează pe activităţile organizaţiei. Ei sprijină această organizaţie cu toată puterea lor, pentru că ei văd un sens doar în acţiuni concentrate , acestea putînd fi pornite doar de către organizaţii mari. În schimb Microsoft din Unterschleissheim nu vrea să renunţe la iniţiativele sale proprii. "În principiu noi acţionăm împotriva oricărui fel de furt de soft", ne asigură Ruth Bachmann, referent de presă al întreprinderii. Microsoft a devenit activ şi împotriva lui PC Data. "Firește că ne coordonăm acţiunile cu iniţiativele VSI". Pentru această coordonare Microsoft a înfiinţat un loc de muncă special. "Ulrike Weinbrenner este specialistul nostru în anti-piraterie, care nu organizează doar conlucrarea cu VSI şi cu ceilalţi producători de soft, ci formulează şi contractele de licenţă şi alte suporturi legale relative la produsele noastre de software", ne mai relatează Ruth Bachmann.

Dacă încercările VSI sau ale producătorilor vor fi încununate cu success, atunci piraţii de soft de la PC Data respectiv Tapol se pot aştepta la o pedeapsă mare. "Deoarece PC Data vinde în mod curent softul copiat ilegal pedeapsa va fi şi mai mare", ne lămureşte procurorul Dieter Wolf. "Dacă eşti prins în particular cu copii pirat, atunci te aşteaptă o pedeapsă de pînă la un an închisoare. În cazul că faci afaceri cu ele, pedeapsa poate ajunge pînă la trei ani închisoare", timp în care poţi să analizezi temeinic dreptatea şi nedreptatea.

(PC Magazin 50/90)

La bunăvoinţa publicului

În numărul 16/90, revista Computer Persönlich din Germania şi-a întrebat cititorii cît de mulţumiţi sînt de hard-ul şi de soft-ul de care dispun. Vă prezentăm în continuare răspunsurile cititorilor nu numai din Germania ci din toată Europa.



Europa a ales, în cazul de faţă nu Parlamentul de la Bruxelles, ci în domeniile hard şi soft. În această privinţă şi-au spus părerea nu doar germani ci şi spanioli, italieni, francezi şi englezi. În multe privinţe se poate vedea unitatea paneuropeană: cam jumătate din cei chestionaţi sînt de părere că sistemul de operare OS/2 nu va deveni niciodată sistemul de operare de bază în întreprinderea lor, deci urmează timpuri grele pentru strategiile pe timp îndelungat ale firmelor IBM şi Microsoft.

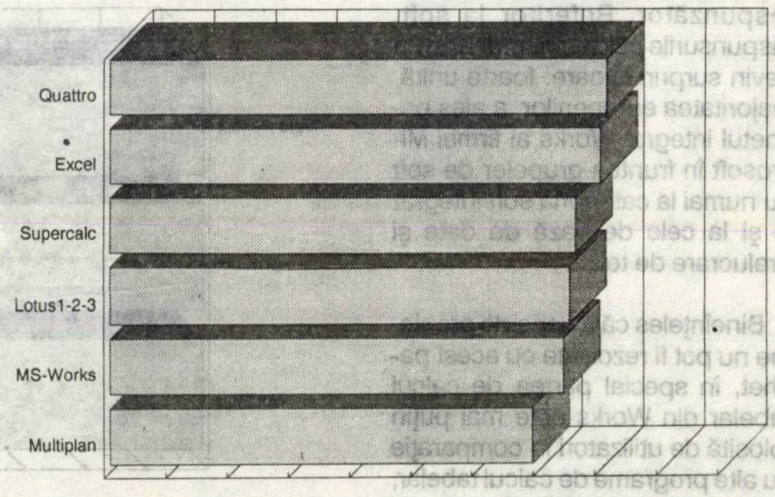
Miezul întregii acţiuni l-a constituit direcţia de sondare a opiniei publice a firmei Datapro din Londra. Sub directa ei conducere a fost conceput un chestionar, tradus în limbile ţărilor respective şi difuzat prin intermediul revistelor de specialitate din cinci ţări europene: "Binary" şi "Datamation" din Spania, "Linea EDP" şi "Linea Personal" din Italia, "01 Informatique" şi "Decision Micro" din Franţa, "Computer Weekly" şi "The Computer Annual Guide to Resources" din Anglia şi de "Computer Persönlich" din Germania.

S-a obţinut un adevărat ,munte de răspunsuri, care odată trimise la Londra au fost introduse în calculator şi au fost prelucrate. Rezultatele vă stau la dispoziţie şi pot fi obţinute de la Datapro sub denumirea de "Annual Computer Users Survey in Europe", pe scurt ACUSE.

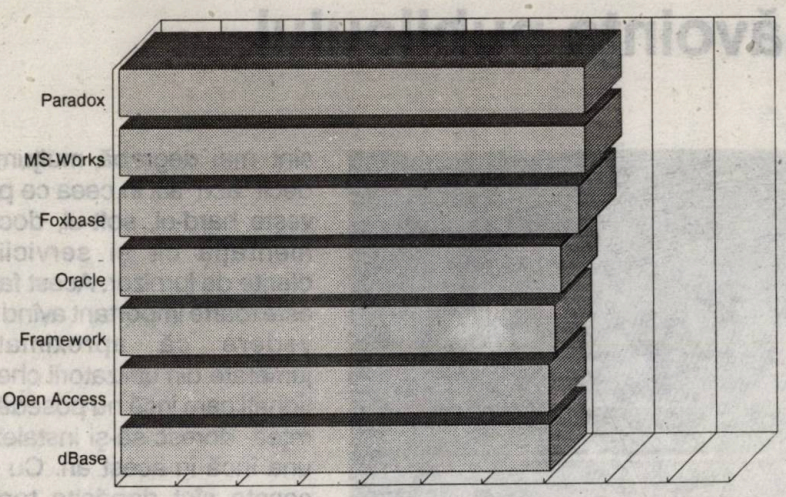
Prima şi cea mai importantă întrebare din domeniul hardware a fost bineînţeles care producător de PC-uri merită cea mai bună notă.

La această probă cîştigătorul este Dell, urmat de Apple şi Compaq. Rezultate bune au obţinut şi Tulip şi HP. Producătorii cel mai slab notaţi sînt Digital Equipement (DEC) în Europa, iar în Germania IBM a fost pe ultimul loc. Mult mai echilibrate sînt rezultatele în ceea ce priveşte producătorii de reţele. Pe primul loc se situează Novell, urmată îndeaproape de IBM-PC-LAN, AppleTalk, 3Com, OS/2-LAN, Torus şi Banyan Vines. În general toţi utilizatorii de reţele de PC-uri sînt mai degrabă mulţumiţi decît "acri" atît în ceea ce priveşte hard-ul, soft-ul, documentaţia cît şi serviciile oferite de furnizor. Acest fapt este foarte important avînd în vedere că aproximativ jumătate din utilizatorii chestionaţi care încă nu posedă o reţea, doresc să-şi instaleze una încă în acest an. Cu aceasta sînt depăşite toate prognozele referitoare la volumul de reţele cerute de piaţă.

Prognozele referitoare la dezvoltarea hardware-ului s-au confirmat în mare parte: dacă în prezent domină PC-urile cu 286, planurile de achiziţionare pentru anul în curs arată în mod cert că 386 intră în vogă, depăşindu-l chiar şi pe 386SX. Dar şi "bătrînele" 286 vor fi mult mai mult utilizate decît "nava amiral" 486. În cazul sistemelor cu 486 se pare că relaţia preţ-performanţe nu a atins încă valori convingătoare. Este preconizată şi achiziţionarea arhitecturilor RISC ale diferiţilor producători, dar acestea joacă un rol tot atît de minor ca şi familia Motorola 68XXX. Această situaţie caracterizează, cu mici excepţii, toată Europa.



Programe de calcul tabelar



Baze de date

Diferenţe mai mari sînt observate în domeniul service-ului. În timp ce pe continent se apelează în mod clar la sprijinul furnizorilor de specialitate, în Anglia locul întîi este ocupat de firmele specializate pe service. Un motiv pentru această stare de fapt este şi legislaţia din Anglia care joacă un rol important. Se pare că britanicii sînt iniţiatorii unei tendinţe care cîştigă tot mai mult teren şi pe continent. Dacă ne referim însă la întreţinerea din partea firmelor de la care a fost cumpărat PC-ul, atunci firmele DEC, Dell şi HP sînt cele care au obţinut cele mai bune note pentru serviciile efectuate.

La fel de importantă ca şi satisfacţia pentru hard-ul şi service-ul oferit este şi utilizarea unui soft corespunzător. Referitor la soft, răspunsurile utilizatorilor de PC-uri devin surprinzătoare: foarte unită, majoritatea europenilor, a ales pachetul integrat Works al firmei Microsoft în fruntea grupelor de soft nu numai la categoria soft integrat ci şi la cele de bază de date şi prelucrare de texte.

Bineînţeles că o serie de probleme nu pot fi rezolvate cu acest pachet, în special partea de calcul tabelar din Works este mai puţin folosită de utilizatori în comparaţie cu alte programe de calcul tabelar, cum ar fi Lotus 1-2-3. Cei care-şi pot rezolva problemele cu Works sînt foarte mulţumiţi însă şi îi dau calificativul clasei medii, alături de Lotus 1-2-3 şi Supercalc.

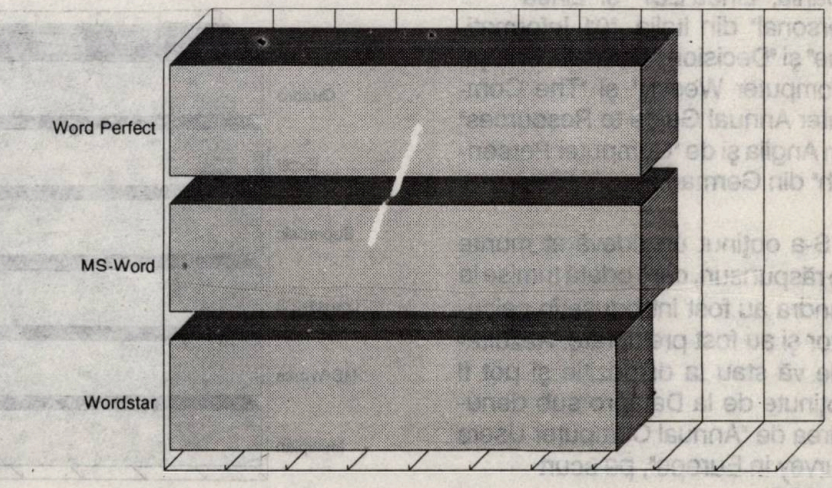
Acesta este un semn pentru producătorii de soft, de a-şi concepe programele lor într-un mod unitar şi uşor de deservit. În acest sens Windows, program unitar cu o suprafaţă uşor de deservit, va putea deveni o nouă unitate de măsură.

Programul de calcul tabelar considerat de utilizatori a fi cel mai bun este programul Quattro Pro al firmei Borland, urmat îndeaproape de programul Excel al firmei Microsoft. Pe de altă parte casa de soft are la această categorie şi programul care se situează pe ultimul loc, este vorba de programul Multiplan, care nu mai este acceptat de utilizatori.

În cazul bazelor de date calităţile preferate sînt deservirea uşoară şi viteza. Nu este de mirare că pe primul loc alături de Works se află şi Paradox, SGBD-ul firmei Borland. Cu Paradox oricine poate construi rapid o bază de date şi o poate administra în mod eficient.

Acelaşi lucru este valabil şi pentru Foxbase, care se află pe locul doi; oricine poate învăţa rapid să lucreze cu ferestrele ecran ale acestuia (care din păcate nu sînt compatibile cu Windows). dBase se află abia după Oracle, Framework şi Open Access. Se vede că un SGBD la care mai întîi trebuia scris un program pentru a putea vedea ceva pe ecran nu mai mulţumeşte astăzi aproape pe nimeni.

Evaluarea programelor de procesare de texte este puţin surprinzătoare: pe locul întîi se află Works împreună cu Wordperfect, cu toate că primul dintre ele este folosit de mai mulţi dintre utilizatorii chestionaţi decît al doilea. Şi în cazul defalcării problemei pe subpuncte: documentaţie, prietenos cu utilizatorul, asistenţă din partea comerciantului şi performanţe, cei doi rivali nu prezintă diferenţe semnificative. Wordstar este folosit cam de acelaşi număr de utilizatori



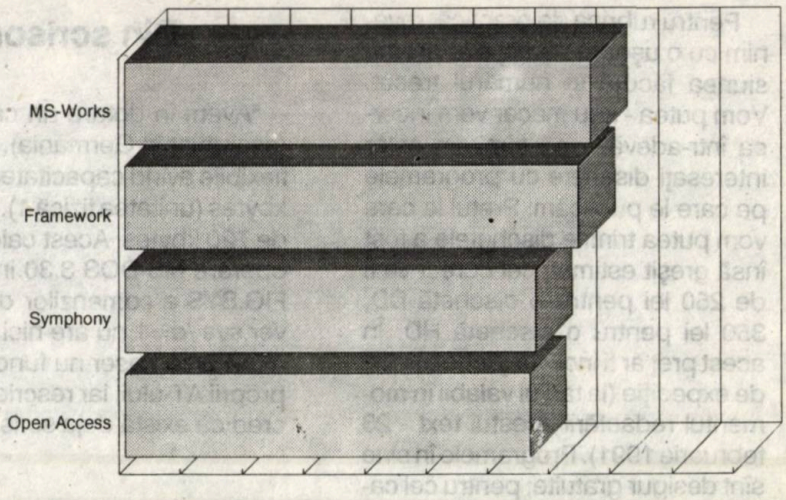
Procesoare de texte

ca şi Wordperfect, dar ca punctaj se clasează la un pas bun de primele două, obţinînd punctaj mai mic mai ales în ceea ce priveşte documentaţia şi performanţele.

În domeniul pachetelor integrate Works conduce la o distanţă relativ mică de Framework. La o distanţă destul de mare de primele două se află Symphony şi Open Access versiunile II şi III.

Cu toate că pachetele integrate au reuşit să pătrundă în domeniul altor programe, domeniul lor principal nu este folosit în întregime, de exemplu un mic grup de utilizatori foloseşte partea de calcul tabelar cuplată cu prelucrarea de texte şi cu conceperea unor baze de date mici.

Interesantă pentru dezvoltarea pe termen lung este întrebarea referitoare la sistemul de operare: cam jumătate din răspunsuri arată că OS/2 nu este considerat c-ar putea deveni sistemul de operare numărul unu în întreprinderile în care lucrează. Doar o cincime din cei chestionaţi afirmă că OS/2 este, sau va deveni în următorii doi ani, sistemul de operare principal în



Pachete integrate

întreprinderea lor, iar un sfert din utilizatori plănuiesc introducerea lui OS/2 în următorii 2-5 ani.

În această privinţă există şi diferenţe interesante: în timp ce doar o treime dintre francezi cred că OS/2 nu va fi niciodată folosit la ei, în Germania şi Anglia cu aproape 60% mai mulţi sînt utilizatorii care cred acest lucru. Aproape o treime din cei chestionaţi în Franţa sînt nedecişi, deci aici numărul celor care încă nu ştiu ce vor este cel mai mare.

Nici Unix nu rămîne neamendat: 50% dintre italieni îl refuză, în media nota sa nu este mai bună decît cea obţinută de OS/2.

În concluzie încă nu s-a cristalizat o imagine clară asupra sistemului de operare care va fi folosit cu precădere în anii 90. Cea mai probabilă va fi coexistenţa paşnică a sistemelor de operare Unix şi OS/2.

(Computer Persönlich 2/91, Ulrich Kruppe)

Staţie de lucru

Aparatul denumit "staţie de lucru" aparţine firmei Fujitsu şi va fi prezentat la tîrgul internaţional CeBIT de la Hanovra. Acesta este un PC 486 cu televizor, cameră video, cartelă fax, staţie telex, scanner, telefon cu răspuns automat, modem, copiator color şi imprimantă laser color, toate integrate. Echipamentul va fi oferit pentru o sumă cuprinsă între 30.000 DM şi 40.000 DM.

Se pune întrebarea: de ce şi-ar mai cumpăra cineva un PC şi pe lîngă el să mai cumpere un copiator, un scanner şi toate celelalte ? Răspunsul îl putem amîna pînă la CeBIT. Atunci cei de la Fujitsu vor pune această întrebare clienţilor lor şi de ce nu şi d-voastră stimaţi cititori.

Calul troian ?

Groază în zona Golfului: În cadrul operaţiunii "scutul deşertului" executată de trupele ONU sub conducerea SUA, timp de trei luni au fost înregistrate cel puţin 5.000 de cazuri de viruşi pe calculatoarele din dotare. Ştirea a fost anunţată de "Governement Computer News".

Programele de sabotare a calculatoarelor au ajuns în PC-uri deoarece soldaţii plictisiţi de căldura deşertului şi-au folosit timpul liber cu încărcarea pe calculatoarele armatei a unor jocuri. S-a ajuns la concluzia că virusul a fost propagat de copiile pirat ale jocului "Solitaire". Comentariul purtătorului de cuvînt al guvernului, Billie Gaskill: "Se aude ticăitul unei bombe cu ceas".

Pentru rubrica de practică, revenim cu o uşoară corecţie la promisiunea făcută în numărul trecut. Vom putea - sau măcar vom încerca într-adevăr - să trimitem celor interesaţi dischete cu programele pe care le publicăm. Preţul la care vom putea trimite dischetele a fost însă greşit estimat; cel corect va fi de 250 lei pentru o dischetă DD, 350 lei pentru o dischetă HD. În acest preţ ar fi incluse şi cheltuielile de expediţie (la tariful valabil în momentul redactării acestui text - 23 februarie 1991). Programele în sine sînt desigur gratuite; pentru cei care ne-ar trimite dischete proprii spre a prelua programele, ar rămîne numai partea de cheltuieli legate de copiere/expediere, estimată la 50 lei/dischetă. O Valoare exactă încă nu o ştim nici noi.

O adăugire relativă la programul mformat publicat în numărul trecut. La primul test cu dischete de 800 kbytes, totul merge bine, dar a doua zi, după ce calculatorul a mai stat o noapte, nu mai vrea sub nici o formă să recunoască discul pe care el singur l-a formatat şi scris cu o zi în urmă! Dacă vi s-a întîmplat aşa ceva, de vină este modificarea permanentă a locaţiei de memorie 0000:0490, făcută de mformat la formatare la 800 kbytes şi lăsată aşa. O soluţie ar putea fi următorul programel, numit 800.com care ar trebui lansat înainte de a trece pe lucrul cu dischete de 800 kbytes. Introducerea se face de la nivel DOS.

```
debug
- a100
xor ax,ax
mov ds,ax
mov byte ptr [0:490],54
mov ax,4c00
int 21

rcx
10
n800.com
w
```

Din scrisorile primite la redacţie

"Avem în dotare un calculator compatibil PC/XT marca GULIP (asamblat în Germania), calculator care nu poate citi decît discuri flexibile avînd capacitatea de 1.2 Mbytes (unitatea fizică 0) sau 360 kbytes (unitatea fizică 1). Aş fi interesat să pot utiliza discuri flexibile de 720 kbytes. Acest calculator ne-a fost livrat doar cu sistemul de operare MS-DOS 3.30 instalat pe harddisk. Menţionarea în CONFIG.SYS a comenzilor device = driver.sys /d=0 şi device = driver.sys /d=1 nu are nici un efect. Driverul ATDQ, preluat de la un AT m,arca Laser nu funcţionează, bineînţeles, folosind întreruperi proprii AT-ului, iar rescrierea unui driver îmi pare inutilă, atît timp cît cred că există deja scris."

Godeanu Oleg
str. Nicolae Bălcescu 39, apt. 5, 6200 Galaţi.

"Cred că ar fi de interes pentru toţi cititorii revistei publicarea unor adrese de reviste de specialitate din străinătate, cît şi adrese ale unor firme de soft şi/sau hard şi ale unor edituri gen McGraw Hill, Osborne etc. ... Am dori să ne ajutaţi cu privire la procesorul cu care este dotat calculatorul nostru. Este vorba de un Western Digital cu codul V40. Comportarea diferitelor tipuri de programe de diagnostic ne-a lăsat "puzzle": unele îl identifică ca fiind 8088, altele ca 8086, ba am avut surpriza să ni se spună că pe placă avem coprocesor 87 (şi nu avem nici măcar soclu pentru aşa ceva!). Bănuim că ar trebui să fie ceva gen '80188', adică un 8088 mai puternic (şi mai rapid) cu coprocesor incorporat, dar consultînd cataloagele **intel** nu am găsit că ei ar fi fabricat ceva de genul acesta. Cupla de magistrală a XT-ului nostru este cu un singur conector, adică la fel ca în calculatoarele cu magistrala de date multiplexată pe 8 biţi. Rugăm pe oricine dispune de foi de catalog, foi de prezentare sau de 'ceva' despre V40 să ne ajute."

Maierean Radu
str. arh. Louis Blank, 2 , bl.I1, sc. A, ap. 12, 71.208 Bucureşti 1

"Cum pot breveta cîteva programe ştiinţifice din domeniul metalurgic, scrise în TP5, TC2? ...Plotterul PIF-03 este compatibil HP ? ... Ştiţi cumva motivele incompatibilităţii parţiale a unităţilor floppy de 5,25" aflate în configuraţia Junior XT86? (erori la citire de pe dischete scrise pe alte calculatoare, nerecunoaşterea dischetelor de 720 kbytes (de la pista 40 în sus)? Toate aceste calculatoare (Junior XT86) au acest defect. Trebuie schimbată unitatea floppy ? ... Ştiţi cumva preţurile de la noi pe piaţa neagră a I8087, 80287? "

Marinescu Ion
str. Scarlat Vîrnav nr. 19, Bucureşi sector 6

Redacţia îşi arogă dreptul de a opera scurtări în text, fără a denatura însă sensul celor comunicate. Luări de poziţie prezentate aparţin semnatarilor lor şi nu reprezintă părerile sau punctul de vedere al redacţiei.

Atlas soft
Înconjurul lumii în 60 de minute

Aşa geografia este o plăcere : Cu "PC Globe" şi cu calculatorul într-o expediţie în jurul lumii.

Aţi ştiut că în Samoa de Vest se plăteşte cu Tala şi că în Caracas tocmai se ia micul dejun, cînd în München se fac pregătirile pentru cafeaua de după amiaza? Probabil că de asemenea este o noutate pentru dumneavoastră faptul că, coordonatele geografice ale Ierusalimului sînt 31,47 grade longitudine nordică şi 35,13 grade latitudine estică şi că Georgetown St. Vincent se află doar la o distanţă de 7626 km de München. V-ar putea interesa de asemenea că suprafaţa ţării Lesotho este de numai 30000 km2 şi că are capitala la Masen.

PC Globe V 3.0 deţine multe informaţii pentru utilizatori. Atlasul de pe dischetă conţine date despre peste 177 ţări şi despre aproximativ 500 dintre cele mai importante oraşe de pe glob.

Dacă Phileas Fogg şi servitorul său Passepartout, eroii lui Jules Verne, au încercat să înconjoare lumea în 80 de zile, puteţi începe o călătorie în jurul lumii cu PC Globe, care costă mult mai puţin şi care se poate termina într-o oră. Începutul este harta lumii. Ea reprezintă punctul de plecare al călătoriei in jurul lumii. Cu ajutorul meniurilor pull-down si al mouse-ului utilizatorul îsi alege un continent, care va apare pe ecran după clic-ul pe mouse. În continuare el poate alege una dintre ţările lumii, cît şi un oraş din cele 500 prezentate. Despre o ţară puteţi afla ciudăţeniile topografice, oraşele cele mai reprezentative sau vîrfurile cele mai înalte ale munţilor. Prin apăsarea unei taste călătorul în jurul lumii poate obţine informaţiile cele mai importante despre ţara pe care abia a "vizitat"-o. Dacă a fost în Gambia, atunci află că limba oficială este engleza şi că apa este potabilă doar în capitala ei, Banjul, că moneda ţării este Dalasi şi că la intrarea în ţară ai nevoie de o dovadă cum că ai fost vaccinat împotriva variolei.

PC Globe nu este un sfătuitor doar pentru turişti ; şi oamenii de afaceri îl pot utiliza pentru a-şi face o părere de ansamblu asupra situaţiei financiare a unei ţări. De exemplu, dacă apeşi o tastă pe mouse aflîndu-te pe Swaziland, şi treci în meniul "bănci de date" afli imediat că Swaziland este o monarhie în care regele Mswati guvernează din 1986. Produsul social brut a fost în 1989 în jurul valorii de 1,088 milioane dolari şi s-a obţinut mai ales din servicii şi agricultură. Mărfurile de export sînt zahăr, azbest, produse lemnoase, fructe citrice şi carne. Doriţi mai multe informaţii? În Swaziland se extrag şi se prelucreaza alături de zinc, cărbune, azbest şi diamante şi aur ca materie primă.

Şi pentru un globetrotter pe care îl interesează problemele sociale PC-Atlas este pregătit.Aşa de exemplu poate afla că în Lesotho bărbaţii şi femeile trăiesc în medie 50 de ani. Populaţia ţării dispune de 88 de spitale, ceea ce ar echivala cu 650 locuitori/pat. În schimb fiecare dintre cei trei stomatologi existenţi trebuie să consulte peste 0,5 milioane de locuitori.

Dacă cineva are o mătuşă în Port Moresby din Papua - Noua Guinee, atunci PC Globe îl va opri prin utilitarul "zone de timp" să o sune la o oră nepotrivită. Nici distanţele nu reprezintă o problemă pentru PC Globe. Din meniul "utilitare" se alege "distanţa dintre oraşe" şi utilizatorul va afla imediat că distanţa dintre Entebbe din Uganda şi München este de 5.706 km. O călătorie mai lungă ar fi la Christchurch în Noua Zeelandă. Dacă punctul de pornire ar fi la München atunci ar fi de parcurs 18.455 km.

La fel de repede programul ne poate informa despre cursul valutar din toată lumea. De exemplu în Lesotho pentru o marcă primeşti 1.50 maloti. În Djibouti primeşti 10.390 djibouti pentru 100 de mărci.

Cine nu se mulţumeşte cu bazele de date pregătite în atlasul calculatorului poate să le întregească singur. Într-un meniu special se pot compara datele diferitelor ţări începînd cu produsul social brut, creşterea numărului populaţiei, statistici de sănătate, pîna la bugetul apărării.

La versiunea 3.0 s-a adăugat programul "PC Nations". Aici se pot vedea steagurile a 175 ţări si asculta imnurile fiecărei ţări în parte. Datele şi ecranele pot fi tipărite şi pe imprimantă. PC Globe permite şi memorarea unor date în fişiere text speciale care pot fi prelucrate ulterior. PC Globe poate fi instalat atît pe hard-disk cît şi pe dischetă, are nevoie de 512 kbyte memorie internă şi de o placă grafică. Preţul atlasului dicţionarului şi ghidului turistic poate ajunge la 170 de mărci.

Chiar şi călători mai puţin versaţi se pot descurca foarte uşor cu PC Globe. Trebuie spus că programul nu poate fi modificat în interese personale, el vrea doar să informeze corect despre ţări şi oameni, dar nu poate fi folosit în lucrări geografice. Pentru ca datele şi informaţiile să nu fie depăşite, în acest an el va fi actualizat cu cele mai recente informaţii despre ţări. Programul este un lucru interesant, care ne aduce lumea mai aproape, fără să fie nevoie să ne luăm după urmele lui Phileas Fogg.

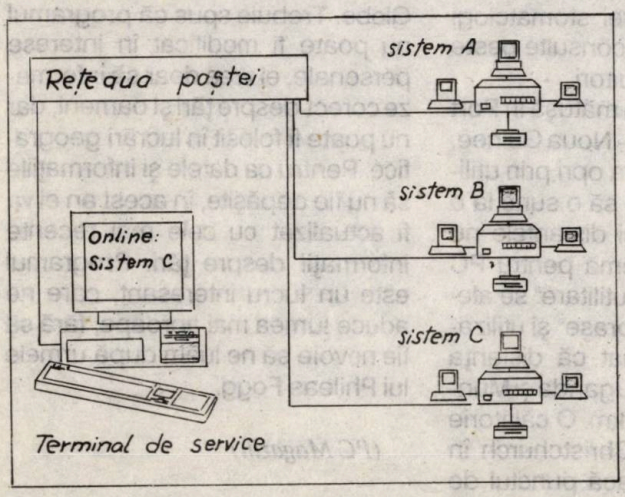
(PC Magazin)

Remote-Control Software

Cu ajutorul aplicaţiilor remote control (de comandă la distanţă) şi cu ajutorul unui cuplor acustic, un utilizator poate lua, cum s-ar zice, un alt PC pe fir. Aceasta nu înseamnă altceva decît faptul că el îl va putea comanda de la distanţă prin intermediul PC-ului propriu.

Un exemplu: Utilizatorul A a primit un program nou, pe care ar dori neapărat să-l arate şi utilizatorului B. Numai că A locuieşte la cîteva sute de kilometri de B. Despre o deplasare pînă la el nici nu poate fi vorba. Şi fiind vorba de software comercial, în mod normal nu dispune de o copie. Cu toate acestea B poate urmări, la el acasă, în mod legal, cum lucrează software-ul respectiv. Condiţia este ca amîndoi să posede acelaşi software de comandă la distanţă. După ce amîndoi şi-au instalat software-ul de comandă la distanţă, utilizatorul A face un apel utilizatorului B prin intermediul modem-ului. Legătura este realizată de software-ul de comandă la distanţă. După scurt timp utilizatorul B vede, ca de obicei, cursorul DOS (de ex. C:\). Din acest moment el poate lucra ca de obicei cu PC-ul. Ciudăţenia o constituie faptul că el nu lucrează cu PC-ul propriu, ci cu PC-ul utilizatorului B. În acest mod poate lansa în execuţie programe, poate executa comenzi DOS, pe scurt: poate folosi toate facilităţile pe care i le oferă PC-ul gazdă. Utilizatorul B vede acum mediul de lucru oferit de PC-ul utilizatorului A. Aceasta înseamnă că toate comenzile pe care le dă B se execută pe PC-ul utilizatorului A, deci pe PC-ul "străin". Dacă este lansat un program în execuţie, el va rula pe PC-ul "străin". Ieşirea spre monitor este comutată pe PC-ul utilizatorului B, astfel încît acesta este folosit exclusiv ca terminal. Această facilitate este asemănătoare cu cea oferită de o reţea. Dacă, de exemplu, nu se posedă decît un XT şi unul din colegi dispune de un 386, atunci de pe XT se poate lansa în execuţie software scris pentru 386, deoarece XT-ul serveşte doar ca terminal şi pe el se afişează doar ceea ce în mod normal ar afişa sistemul cu 386.

Întrucît controlul asupra PC-ului "străin" este total, nu mai există nici o deosebire între a accesa direct un PC prin intermediul unui software de comandă la distanţă, şi a sta în faţa PC-ului respectiv. În acest mod A poate lucra cu programul, cu toate că nu îl are instalat pe calculatorul propriu, normal că aceasta durează numai atît timp cît este realizată legătura, pentru că în cele din urmă el foloseşte doar ieşirea la terminal şi intrarea de la tastatură şi nu are programul transferat în calculatorul propriu.



Pentru firmele de software se deschid astfel noi modalităţi de prezentare a produselor. Însă nu numai programele sînt cele care pot fi utilizate în acest mod. Întrucît controlul asupra PC-ului "străin" este total se poate utiliza şi periferia acestuia. În acest mod un utilizator din Baia Mare, de ex., poate lista un fişier pe imprimanta unui coleg din Constanţa. Există însă şi posibilitatea de a utiliza propria imprimantă. Să presupunem că utilizatorul din Constanţa trebuie să realizeze un grafic de prezentare şi că nu dispune de soft-ul necesar. Partenerul de afaceri din Baia Mare însă posedă acest soft. În acest caz din Constanţa poate fi utilizat programul colegului din Baia Mare pentru realizarea graficului respectiv. Apoi în mod natural graficul trebuie imprimat. Dacă listarea se face pe imprimanta colegului din Baia Mare, deoarece pe acesta se găseşte soft-ul, colegul de acolo va trebui să trimită graficul la Constanţa prin intermediul poştei. Cu ajutorul soft-ului de comandă la distanţă acest lucru nu mai este necesar, deoarece listarea poate fi comutată direct pe imprimanta proprie, din Constanţa în cazul nostru.

Un alt exemplu practic este din domeniul serviciilor. Să presupunem că un utilizator şi-a cumpărat recent un harddisk, pe care nu ştie să-l formateze deoarece nu a mai lucrat cu utilitare de formatare. Prin intermediul unui soft de comandă la distanţă, un coleg poate prelua operaţia respectivă şi poate dichisi discul.

Pentru domeniul serviciilor apar astfel noi posibilităţi. Tehnicienii vor putea rezolva în viitor astfel de probleme direct din cadrul unităţii, economisind astfel timp şi bani. Parţial, aceste modalităţi au şi început să fie aplicate în practică (normal că nu la noi).

Şi încă un cuvînt despre cerinţele impuse de utilizarea unui soft de comandă la distanţă. Este necesar un PC cu o configuraţie standard şi un modem standard. Pentru lucrul profesional se recomandă un modem de viteză înaltă,

necesar mai ales pentru programele care lucrează în mod grafic.

Carbon Copy Plus

Unul din cele mai cunoscute programe de comandă la distanţă este programul Carbon Copy Plus. Acesta se livrează în două versiuni, pentru calculatorul apelant şi pentru cel de apelat.

Programul pentru calculatorul de apelat este mai scurt şi se încarcă rezident în memorie, necesitînd 62 Kbytes. Din păcate nu există o opţiune de plasare a lui în memoria extinsă sau expandată. Comenzile programului se transmit prin intermediul unor combinaţii de taste, neexistînd, din păcate, posibilitatea de a opera cu un mouse. Cu toate acestea operarea este transparentă şi simplă.

Carbon Copy Plus

Furnitura de livrare:
3 dischete, un manual

Cerinţe sistem: MS-DOS de la 3.1 în sus, host 200 Kbytes memorie de lucru, slave 65 Kbytes memorie de lucru, XT, AT, 386, sau 486.

Operare: tastatură

Producător/ofertant:
dr. Neuhaus, Hamburg

Preţ (inclusiv impozitul pe valoarea adăugată):
aprox. 900 DM

O funcţie utilă este, cu siguranţă, funcţia anterior descrisă, de comutare a listării pe imprimanta proprie. În afară de aceasta programul permite şi transmiterea unor fişiere la distanţă. Este posibil şi un "teledialog scris". Programul mai are implementat şi un emulator de terminal, astfel încît prin intermediul lui pot fi folosite şi programe convenţionale de transmitere a datelor la distanţă (de exemplu selecţia unei cutii poştale).

Instalarea şi configurarea programului sînt lipsite de probleme. Programul recunoaşte un număr foarte mare de modemuri, şi nu are nici un fel de probleme cu transmiterea graficelor, putînd transmite pînă şi grafice VGA.

Co/Session

Din punct de vedere optic, Co/Session este foarte asemănător cu concurentul său Carbon Copy Plus. Operarea se face tot de la tastatură, nu poate fi utilizat un mouse, se livrează într-o variantă pentru calculatorul apelant şi într-alta pentru cel de apelat.

Programul pentru calculatorul de apelat se încarcă tot rezident în memorie, dar necesită 70 Kbytes. Printre funcţii se numără un emulator de terminal, o carte telefonică şi posibilitatea de a comuta listarea pe imprimanta proprie. De asemenea sînt posibile transmiterea de fişiere şi teledialogul. În practică însă au apărut probleme la transmiterea graficelor VGA, atunci cînd cele două PC-uri posedau plăci grafice diferite. În rest şi Co/Session s-a dovedit a fi un utilitar de nădejde. La transferul în mod text (de ex. din Word, PC - Tools, etc.) n-au fost probleme. N-au fost probleme nici în cazul graficelor simple, de ex. în mod CGA. Au apărut unele probleme şi la lucrul cu modemurile de tip MNP. Pe ansamblu însă programul este bun, uşor de învăţat şi simplu de utilizat.

(Computer Persönlich 1/91, Lars Blumenhofer)

Co/Session

Furnitura de livrare:
două dischete, un manual

Cerinţe sistem: MS-DOS de la 3.2 în sus, host 165 Kbytes memorie de lucru, slave 70 Kbytes memorie de lucru, XT, AT, 386 sau 486 cu harddisk.

Operare: tastatură

Producător/ofertant:
TTC, Berlin

Preţ (inclusiv impozitul pe valoarea adăugată):
aprox. 796 DM

Tabel comparativ

Nume	Carbon Copy Plus	Co/Session
Preţ aprox. (DM)	900	796
Configuraţie		
Descărcabil din memorie	nu	da
Suport Hayes	da	da
Suport MNP	da	nu
Suport PEP	nu	nu
Modemuri predefinite	da	da
Comutare imprimantă	da	da
Posibilităţi Chat	da	da
Transfer de grafică	da	da
Transfer de texte (inclusiv culoare)	da	da
Operare		
Mouse	nu	nu
Meniuri	da	da
SAA	nu	nu
Emulare terminal	da	da
Carte de telefon	da	da
Transfer fişiere	da	da
Funcţie Help	nu	nu
Limbaj de programare	nu	da

Calcul tabelar

Sub denumirea de prelucrare de texte fiecare îşi poate imagina ceva. Nici denumirea de program de grafică nu ascunde o enigmă. Dar cea de calcul tabelar (spreadsheet) ? Această categorie de software nu se mai autodefineşte tot la fel de simplu încă din denumire. Aceasta se explică prin faptul că apariţia acestei "unelte" a fost posibilă abia după apariţia calculatoarelor personale.

Un procesor de texte nu este altceva decît o maşină de scris mai bună. Un program grafic este o planşetă de desen electronică. Noţiunea de calcul tabelar este mai abstractă. Aceasta este şi explicaţia pentru care răspîndirea acestei categorii de programe este mai redusă decît cea a procesoarelor de texte. Un începător nu ştie ce-ar putea să înceapă cu o astfel de unealtă. Un program de calcul tabelar poate însă rezolva multe probleme - şi nu numai din cele pe care nu le-aţi avut înainte.

Un program de calcul tabelar poate fi imaginat în modul următor: o foaie mare de hîrtie împărţită în linii şi coloane. Această foaie de hîrtie este prevăzută cu un sistem de coordonate. Coloanele sînt notate cu litere, iar liniile cu cifre arabe. Intersecţiile liniilor cu coloanele sînt denumite celule. Cu ajutorul acestui sistem de coordonate o celulă este determinată în mod unic. Astfel celula din colţul din stînga sus se numeşte A1, cea de lîngă ea B1, iar cele care se află sub ele se numesc A2 şi respectiv B2. Deci asemănător cu o tabelă de şah numai mult mai mare.

Fiecare celulă poate conţine trei tipuri diferite de date: un text, un număr sau o "formulă". Ultimul tip este găselniţa întregii probleme.

Pentru un program de calcul tabelar, o formulă este o instrucţiune care operează cu conţinutul uneia sau mai multor celule. Un exemplu: Să presupunem că în celulele A1 pînă la A5 se află numere. Se poate scrie o formulă care să adune cele cinci numere şi care să fie plasată, de ex., în celula A7. În această celulă trebuie scrisă, deci, formula: "A1+A2+A3+A4+A5". Nici nu s-a terminat bine de scris formula că în celula A7 şi apare rezultatul.

Toate bune şi frumoase, numai că în cazul în care trebuie adunată o coloană lungă de cifre, formula devine şi ea foarte lungă. Pentru a evita aceasta se poate lucra cu aşa-numitele funcţii. Acestea acţionează asupra unor domenii întregi, funcţiei trebuind să i se specifice doar primul şi ultimul argument.

În cazul celui mai cunoscut program de calcul tabelar, Lotus 1-2-3, în cazul exemplului nostru ar trebui scris: "@SUM (A1..A5)". La un alt program, cum ar fi Excel, ar trebui scris: "SUM (A1:A5)". Bineînţeles că domeniul poate să cuprindă şi mai multe coloane. De ex. celulele A1 şi D5. Trebuie ştiut doar că prima şi ultima coloană determină un domeniu dreptunghiular.

După cum se poate observa sintaxa se formează de fiecare dată după aceaşi schemă: o instrucţiune (@SUM, sau SUM) şi un domeniu (A1..A5, sau A1:A5). Pe scurt: formează suma conţinutului celulelor de la A1 pînă la A5. Rezultatul va apare instantaneu în celula în care se află formula, în acest caz A7.

Doar adunînd însă nişte numere cu un program de calcul tabelar, ar însemna să-l folosim mult sub capacităţi. Un astfel de program oferă în medie 80 de funcţii, atît matematice cît şi financiare şi statistice. Acestea pot fi interconectate după toate regulile matematicii. Pe scurt, nu există aproape nimic ce să nu poată fi calculat cu un program de calcul tabelar.

Admiţînd că tot ceea ce s-a enumerat aici se poate face şi cu un creion, o foaie de hîrtie şi un calculator de buzunar, numai că mult mai încet, nu este de mirare că un program de calcul tabelar ştie să facă şi multe alte lucruri.

De ex. calcularea mai multor coloane de cifre. Să presupunem că într-o fişă de lucru se află 12 coloane, umplute cu cifre. De ex. încasările unei firme, fiecare coloană reprezentînd o lună a anului. Ar fi destul de obositor să scriem de 12 ori aceeaşi formulă pentru a aduna cifrele de pe coloane.

Programele de calcul tabelar oferă o soluţie mai simplă: formula se scrie o singură dată şi se copiază de un număr corespunzător de ori în coloanele rămase. De efect este faptul că toate referinţele se actualizează în mod automat. Aceasta înseamnă că dacă formula de însumare a primei coloane, de ex., @SUM(A1..A10) s-a copiat în coloana a doua ea se va transforma automat în @SUM(B1..B10).

Se poate spune că adresele domeniului (numele celulelor) sînt interpretate relativ la celula care conţine formula., sau astfel spus formula nu spune în acest caz "adună celulele A1 pînă la A10" ci "adună cele zece coloane de deasupra mea". Dacă se doreşte, însă, referinţele pot fi şi absolute.

O altă caracteristică ridică şi mai clar un program de calcul tabelar deasupra creionului, foii de hîrtie şi calculatorului de buzunar. Dacă se schimbă doar o singură cifră, întreaga tabelă este recalculată instantaneu. O adevărată binecuvîntare pentru orice comerciant care calculează costurile unui produs. El poate vedea imediat la ce preţ poate obţine un profit.

Multe programe de calcul tabelar mai fac încă un pas şi oferă funcţii pentru analize de genul "ce s-ar întîmpla dacă". Deciziile devin astfel o joacă de copii.

Programele de calcul tabelar mai oferă însă şi altceva, şi anume aşa-numitele macrouri. Prin "macrou" se înţelege o suită de comenzi care se pot executa printr-o singură apăsare de tastă. În acest mod se poate economisi multă muncă de rutină, deoarece comenzile care se repetă foarte des se pot reduce la o singură comandă.

Cele mai multe programe merg şi mai departe, posedînd aşa-numite înregistratoare de macrouri. La o anumită comandă acesta este cuplat şi toate intrările de la tastatură vor fi înregistrate. După ce modul de lucru dorit a fost înregistrat în întregime, înregistratorul se decuplează printr-o altă comandă şi macroul este gata. Acesta poate fi ulterior editat pentru a face modificări sau pentru a-l adapta altor medii de lucru.

Programele de calcul tabelar bune posedă şi un limbaj de programare a macrourilor, cu ajutorul căruia se poate programa efectiv. Acestea dispun atît de instrucţiuni repetitive cît şi de posibilităţi de ramificaţie la alte macrouri sau submacrouri. Pentru Lotus 1-2-3, standard-ul programelor de calcul tabelar, este oferit spre vînzare chiar şi un compilator de macrouri, care oferă o paletă de comenzi mai bogată şi măreşte viteza de execuţie.

Proiectanţii profesionişti de software utilizează limbajele moderne de programare a macrourilor pentru realizarea unor aplicaţii întregi. Cîte un contabil nici nu ştie că în principiu el lucrează cu un program de calcul tabelar.

De cînd programul Lotus 1-2-3 se găseşte pe piaţă, (şi aceasta reprezintă în era calculatoarelor o jumătate de eternitate), cele mai multe programe de calcul tabelar se compun din trei părţi: calculul tabelar, o bază de date şi un modul grafic. Din acest motiv firma din SUA, Lotus Developement, şi-a denumit produsul Lotus 1-2-3.

La majoritatea programelor, bazele de date nu sînt cine ştie ce. Bazîndu-se în fapt pe calculul tabelar, puterea lor este limitată, dar oricum pentru majoritatea aplicaţiilor ar trebui să fie suficientă.

Cu totul diferit se prezintă situaţia în cazul modulelor grafice. În acest domeniu, majoritatea programelor de calcul tabelar oferă facilităţi care le-ar prinde bine chiar şi programelor de grafică economică. Acestea ajung de la diferitele tipuri de grafice, cum ar fi diagrame tip tort sau histogramă, pînă la diferite tipuri de scriere ce pot fi listate într-o calitate de prim rang. În acest mod cifrele calculate pot fi prezentate într-o formă mult mai intuitivă.

Lotus 1-2-3 2.2

De cînd există calculatoarele personale, Lotus 1-2-3 este standardul pentru programele de calcul tabelar. Acest program a ajutat la punerea pe picioare a unei întregi industrii. Nenumărate case de soft şi-au început cariera programînd clone de Lotus 1-2-3.

Suprafaţa 1-2-3, limbajul de programare al macrourilor ca şi setul de comenzi s-au impus ca o cerinţă de neînlocuit pentru majoritatea programelor de calcul tabelar.

Cu versiunea 2.2, apărută în urmă cu aproape un an, Lotus a încercat să adapteze versiunea iniţială stadiului general de performanţe. Bineînţeles că s-a mers numai atît de departe încît programul să poată fi folosit chiar şi cu o configuraţie hard "spartană". Şi pentru această versiune este suficient un PC cu două unităţi floppy şi 320 Kbytes memorie.

În acest caz însă nu vom putea savura facilităţile oferite de Allways. Acest aşa zis "add-in" (amănunte în continuare) oferă utilizatorului de programe de calcul tabelar facilităţi DTP (DeskTop Publishing) necunoscute pînă în prezent. Astfel pare logic ca Lotus Devellopement să recomande un harddisk şi cel puţin 512 Kbytes RAM atunci cînd 1-2-3 se foloseşte împreună cu Allways. Experienţa arată însă că pot apare probleme şi cu 640 Kbytes RAM.

Cu toate acestea pentru utilizatorii de Lotus 1-2-3, Allways este o revelaţie. Dintr-o dată la dispoziţia lor se află mai multe tipuri de scriere, de o calitate deosebită şi la diferite dimensiuni. Pe aceeaşi pagină pot fi tipărite atît grafice cît şi tabele, iar liniile şi chenarele pot spori efectul prezentării.

Cel ce nu cunoaşte 1-2-3 va avea probleme la început. Imposibilitatea de a opera cu mouse, suprafaţa orientată caracter şi meniul iniţial invizibil produc greutăţi neiniţiaţilor. Abia după apăsarea tas-

Lotus 1-2-3 2.2

Furnitura de livrare: 12 dischete de 5,25", 360 Kbytes/6 dischete de 3,5", 720 Kbytes/ 2 manuale de cca. 300 şi respectiv 150 de pagini, 3 broşuri, şabloane pentru tastatură.

Cerinţe sistem: MS-DOS de la 2.0 în sus/ 320 Kbytes memorie de lucru, 512 Kbytes pentru lucrul cu

Allways/ 2 unităţi floppy/ XT, AT, 386 sau 486.

Operare: tastatură

Producător/ofertant:
Lotus Developement, München

Preţ (inclusiv impozitul pe valoarea adăugată) în jur de 1700 DM, versiunea pentru reţea (server) cca. 1900 DM, fiecare loc de muncă suplimentar cca. 1000 DM

tei "CapsLock" la marginea superioară a ecranului apare o linie de meniu. Această tastă de apel este valabilă numai pentru versiunea germană, versiunea originală foloseşte tasta "\" (backslash).

Ceea ce pentru începători se dovedeşte a fi un dezavantaj, pentru utilizatorii avansaţi este un avantaj. Ei cunosc deja fiecare combinaţie de taste şi pot lucra aşa mult mai repede. Partizanii continuităţii nu vor avea nimic de reproşat.

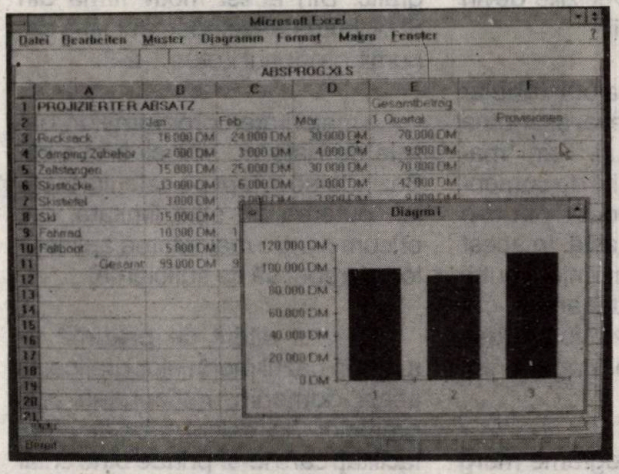
În comparaţie cu concurenţa modernă, pentru vedeta programelor de calcul tabelar de odinioară, prezentul este destul de sumbru. Abia de găseşti o ramură în care Lotus 1-2-3 2.2 să nu se afle la coadă, neluînd în considerare Starplaner, care de fapt este un clon al predecesorului lui 2.2.

Pe de altă parte nu prea există funcţiuni care să-i lipsească; ceea ce i-a lipsit originalului a fost adăugat la versiunea 2.2. Acestea merg de la mijloacele primitive de ajutor, cum ar fi căutarea unor cuvinte cheie într-o tabelă, pînă la comanda UNDO şi la aşa-numita interconectare a fişierelor. Această funcţie permite citirea în tabelă a unor date aflate în fişiere externe. Trebuie spus însă că Excel, QuattroPro şi Lotus 1-2-3 3.1 au mult mai multe de oferit în această privinţă.

Funcţiile macro au fost aduse la cel mai înalt nivel al tehnicii. În această versiune există un înregistrator de macrouri, un manager de macrouri, care poate folosi macrouri externe, şi a fost extinsă şi paleta de funcţii macro.

Unii utilizatori de 1-2-3 ar putea obiecta acum că funcţiile descrise mai sus au existat deja în versiunea 2.01, de ex. înregistratorul. În principiu acest lucru este adevărat, numai că acesta a existat doar ca modul suplimentar. Este vorba în acest caz de mai sus amintitele Add-in. Acestea sînt părţi de program care pot fi adăugate programului principal prin execuţia unei comenzi.

Excel 2.1



Din momentul în care Microsoft a scos pe piaţă Excel pentru Apple Macintosh, acesta a devenit imediat standardul. Versiunea PC are o misiune foarte grea, deoarece Lotus 1-2-3 nu se lasă dat la o parte cu una cu două.

Dar Excel are perspective tot mai bune. În primul rînd pentru că este singurul program de calcul tabelar care se poate cumpăra pentru Windows (rivalul său Wingz, al lui Informix, tot se mai află încă în faşă) şi în al doilea rînd pentru că oferă performanţe corespunzătoare.

Bineînţeles că programul profită enorm de pe urma mediului Windows, atît la operare cît şi în ceea ce priveşte periferia. Ne referim la imprimante, monitoare şi alte echipamente de ieşire. Dacă posedăm hard-ul corespunzător, cu driver-ele Windows potrivite, atunci nu există greutăţi în folosirea lui Excel pe monitoarele de o pagină sau de două pagini. Astfel se pot vedea chiar şi tabele foarte mari dintr-o privire.

Fiind un program tipic Windows, Excel este orientat în mare măsură pe folosirea mouse-ului, cu toate că Microsoft şi-a dat silinţa să nu-i dezamăgească nici pe prietenii tastaturii. Deservirea cu mouse-ul este totuşi mai confortabilă. Pentru marcarea unui domeniu este suficient un clic pe mouse în prima celulă şi deja marcajul poate fi tras peste domeniul dorit. Şi răsfoitul ca şi scroll-ul în foaia de lucru se pot face foarte uşor cu ajutorul mouse-ului. Ei bine, pînă aici şi Quattro Pro 2.0 al firmei Borland, poate ţine pasul, dar faptul că în cadrul programului Excel nici o formulă, respectiv funcţie, nu trebuie introdusă manual face acest program unic.

Ajunge să activezi cu mouse-ul meniul "formule". Un clic scurt pe "introducere funcţii" face să apară o fereastră cu toate cele 129 de funcţii pe care le oferă Excel. După ce s-a ales formula, ea va apare pe loc în celula activă în momentul respectiv. În plus cursorul se va găsi în interiorul celulei între două paranteze, imediat după numele funcţiei. Deci argumentul funcţiei se poate introduce printr-un simplu clic pe mouse, pentru o celulă sau pentru un domeniu de celule. Mai simplu decît aşa nici nu se poate.

Excel 2.1

Furnitura de livrare: 5 dischete de 5,25", 1,2 MBytes/ manual cca. 300 pagini/ şabloane de tastatură.

Cerinţe sistem: MS-DOS de la 3.0 în sus/ Windows 3.0/ 640 Kbytes memorie de lucru/ hard-disk/ AT, 386 sau 486

Operare: tastatură, mouse

Producător/ofertant: Microsoft, Unterschleissheim

Preţ (inclusiv impozitul pe valoarea adăugată): cca. 1700 DM, reţea (5 posturi de lucru) cca. 7500 DM, fiecare post de lucru suplimentar cca. 850 DM.

Acesta a fost doar un exemplu dintre multe care se pot da. Acest mod simplu de deservire este specific întregului program. Puţine programe de calcul tabelar oferă atît de mult confort şi sînt aşa de uşor de învăţat ca şi Excel. Helpurile sînt exemplare iar programul de învăţare este conceput foarte didactic.

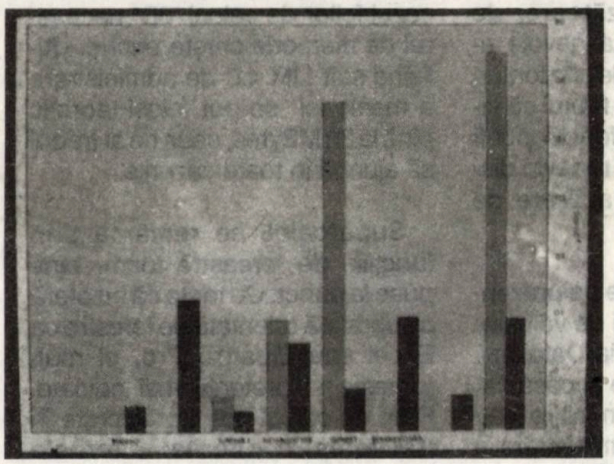
Bineînţeles că şi Excel are şi dezavantaje, nu numai avantaje. Utilizatorii ordonaţi sînt deranjaţi de faptul că după pornire pe ecran apare o foaie de lucru goală, chiar dacă a fost încărcat un fişier nou. Dacă nu o închidem atunci undeva pe ecran vom avea mereu o foaie de lucru goală. Mai grav este faptul că tabelele şi graficele trebuie ţinute pe foi de lucru distincte şi că nu pot fi tipărite pe o singură pagină. La fel de supărător este faptul că în Excel pot fi folosite concomitent doar patru fonturi, cu toate că Windows administrează oricîte fonturi.

În schimb macrourile sînt exemplare. Utilizatorii care nu doresc să programeze pot folosi înregistratorul de macrouri. Avantajos la Excel, în această privinţă, este faptul că macrourile se află pe o foaie de lucru distinctă iar atunci cînd sînt încărcate pot fi folosite de orice fişier activ în acel moment. Din teste a rezultat că limbajul de macrouri este unul din atuurile cele mai puternice ale lui Excel. Volumul limbajului este uriaş şi nu rămîne cu nimic în urma altor limbaje de programare, mai ales că se pot construi nu doar meniuri proprii ci şi boxe de dialog.

Şi modulul de grafică este plăcut. Acesta pune la dispoziţia utilizatorului atît diferite tipuri de grafice cît şi diferite caracteristici de prezentare. Modulul de bază de date se evidenţiază mai ales în legătură cu limbajul de programare a macrourilor şi cu posibilităţile de interconectare a tabelelor.

O problemă mai rămîne totuşi. Puterea reală a programului iese în evidenţă abia pe un AT rapid, sau şi mai bine pe un 386 cu memorie de lucru mare. Dar în acest caz nu programul este de vină ci Windows.

Lotus 1-2-3 3.1



Pe piaţă, în acest moment, se află două versiuni ale lui Lotus pentru PC-uri: versiunea 2.2 şi ultrarecenta versiune 3.1. Dar aceasta din urmă nu trebuie privită ca un urmaş al versiunii 2.2; ambele programe există ca produse de sine stătătoare.

Unul din motivele de bază, care a dus la această situaţie, este imensul suport hard de care are nevoie versiunea 3.1. Ea poate lucra doar pe cel puţin un AT (286) cu 1 MByte memorie de lucru şi cu harddisk. Pe un astfel de suport 1-2-3 3.1 ocupă 5 MBytes. Deci ea nu poate fi folosită pe XT-uri sau pe Laptop-uri. Pentru acestea există versiunea 2.2.

Produsul foloseşte modul de lucru "protected mode" al procesorului. Aceasta înseamnă că procesoarele 286 sînt folosite pînă la ultimul bit, ceea ce nu era cazul în modul de lucru uzual "real mode". De acest lucru era într-adevăr nevoie.

Singurul produs, din acest test, care oferă foi de calcul tridimensionale este Lotus 1-2-3 3.1. În acest caz nu mai avem de-a face doar cu linii şi coloane ci şi cu "foi" suplimentare. Avantajul devine evident. Astfel se poate concepe pentru fiecare lună cîte o foaie separată, pentru a cuprinde, de ex., bilanţurile lunare, adăugînd la sfîrşit o foaie în faţă în care prin însumarea valorilor lunare să se obţină bilanţul anual.

Totul se poate rezolva aşa de simplu deoarece fiecare celulă are acces la celula dinaintea şi respectiv dinapoia sa. O foaie de calcul Lotus 1-2-3 3.1 trebuie imaginată ca un paralelipiped sau ca un teanc de foi. Devine clar de ce sînt necesare cerinţe hard atît de înalte. Un fişier Lotus 1-2-3 3.1 poate cuprinde 8192 de rînduri, 256 de coloane şi 256 de foi. Rezultă în mod pur teoretic un număr de 536.870.912 de celule. Aceasta doar teoretic deoarece dacă în fiecare celulă s-ar afla un număr ar fi necesar un spaţiu liber în memorie de cel puţin 2 GBytes.

Lotus 1-2-3 3.1

Furnitura de livrare: 6 dischete de 5,25", 1,2 MBytes/ 9 dischete de 3,5", 720 Kbytes/ 3 manuale de cca. 350 de pagini şi respectiv 100 de pagini/ 3 broşuri/ şabloane de tastatură

Cerinţe sistem: MS-DOS de la 3.0 în sus/ 1 MByte memorie de lucru/ harddisk/ AT 386 sau 486

Operare: tastatură, mouse

Producător/ofertant: Lotus Developement, München

Preţ (inclusiv impozitul pe valoarea adăugată): cca. 2050 DM, versiunea reţea (server) cca. 2160 DM, fiecare loc de muncă suplimentar cca. 1300 DM.

Pe lîngă foile de calcul tridimensionale Lotus 1-2-3 3.1 mai oferă şi alte "bunătăţi", modulul de grafică de exemplu este excepţional. Cu ajutorul lui pot fi realizate uşor chiar şi grafice tridimensionale.

Şi modul de concepere a tabelelor a fost îmbunătăţit. Astfel avem la dispoziţie mai multe fonturi, cu atributele corespunzătoare, de mărime selectabilă. La nevoie acestea pot fi înlocuite de alte fonturi bitstream. Practic este faptul că diferitele combinaţii de fonturi pot fi depuse în biblioteci şi redevin disponibile la o simplă apăsare de tastă.

În plus Lotus mai are de prezentat şi alte caracteristici de valoare. De exemplu tehnologia Datalens. Datorită ei Lotus poate accesa baze de date externe în citire sau scriere. Partea de administrare a bazei de date excelează prin multitudinea de facilităţi oferite. Cuplînd această facilitate cu limbajul de programare a macrourilor se pot elabora aplicaţii performante.

Dar (pentru că s-a dorit păstrarea compatibilităţii cu versiunile anterioare...) versiunea 3.1 are şi dezavantaje. Acestea ar fi suprafaţa utilizator "retro" şi structura meniurilor. Cine a mai lucrat cu Lotus se poate descurca încă de la pornire ceea ce este un avantaj. Dezavantajoasă este însă ramificarea mare şi în parte nelogică a meniurilor. De vină sînt multele puncte din meniuri care trebuiau adăugate fără a schimba structura lor iniţială. Argumentul pentru 3.1 este însă tridimensionalitatea.

Supercalc5

În timp ce Lotus îşi etalează versiunea 3.1 în reclame şi articole în revistele de specialitate, Supercalc5 nu prea face vîlvă.

Cu toate că acest program oferă ceva foarte asemănător, structura foilor sale de calcul nu este chiar tridimensională. Fireşte utilizatorul nu observă acest lucru. Prin suprapunerea a două foi de calcul bidimensionale şi înmagazinarea lor într-un singur fişier, apare pînă la urmă un mediu tridimensional. Avantajul lui în faţa lui Lotus 1-2-3 3.1 este că poate funcţiona chiar şi cu o memorire de lucru de 512 Kbytes. Dar lucrul devine cu atît mai eficient cu cît avem mai multă memorie la dispoziţie. Mai ales în cazul foilor de calcul "3D" necesarul de memorie creşte enorm. Utilizînd soft LIM 4.0 de administrare a memoriei se pot folosi teoretic pînă la 32 MBytes, ceea ce ar trebui să ajungă în toate cazurile.

Supercalc5 se remarcă prin funcţiile de fereastră foarte bine puse la punct. Cu toate că nu oferă o suprafaţă orientată pe ferestre ca Excel sau Quattro Pro, el mulţumeşte şi pretenţii mai ridicate. Foile de calcul pot fi prelucrate în trei ferestre diferite. Diferitele secţiuni ale unuia şi aceluiaşi fişier pot fi de asemenea prelucrate în ferestre diferite. Pe scurt toate informaţiile necesare la un moment dat pot fi vizualizate concomitent.

Aptitudinile practice ale lui Supercalc5 plac în general. Aşa, de ex., există o opţiune care permite vizualizarea tuturor acelor celule care conţin formule, sau al căror conţinut este dependent de formule sau date din alte celule. În acest mod pot fi depistate foarte uşor referinţele eronate. În plus fiecare celulă poate fi prevăzută cu un comentariu.

Şi programatorilor de macrouri li se oferă posibilităţi remarcabile de depistare a erorilor şi de testare. Astfel pot fi fixate puncte de oprire la dorinţă şi se poate face rulare pas cu pas, ceea ce aminteşte de depanatoarele limbajelor de programare evoluate. Există suficiente posibilităţi de analiză pentru a putea depista orice eroare.

Uimitoare este deplina compatibilitate cu Lotus 1-2-3 pînă în versiunea 2.01. Astfel se pot citi şi scrie fişiere ale versiunilor 1-2-3 1.X (cu extensia .WKS) cît şi 2.01 (cu extensia .WK1).

Chiar şi macrourile Lotus pot fi executate sub Supercalc5 fără probleme. În plus, simultan pot fi folosite atît comenzi Supercalc cît şi Lotus. Posibilitatea de comutare pe structura de meniuri a lui Lotus pune capac la toate. Rezultatul: compatibilitate 100%.

Dezavantajos este faptul că tabelele şi graficele nu pot fi tipărite pe aceeaşi pagină. Supercalc5 are probleme şi cu tipărirea tabelelor în format transversal. Pentru acestea trebuie folosit utilitarul Sideways livrat împreună cu acesta.

În timp ce performanţele tehnice ale lui Supercalc5 conving, deservirea sa lasă mult de dorit. Cu toate că utilizatorul poate alege între structura de meniuri a lui Lotus şi cea proprie, acestea nu sînt cele mai bune.

Şi în acest caz, din dorinţa păstrării compatibilităţii cu versiunile anterioare, funcţiile noi au fost înglobate în structura existentă, fără a se ţine cont de pierderile

Supercalc5

Furnitura de livrare: 9 dischete 5,25", 360 Kbytes/ 2 manuale cca. 300 pag. şi 200 pag./ 2 broşuri/ şabloane de tastatură

Cerinţe sistem: MS-DOS de la 3.0 în sus/ 512 Kbytes memorie de lucru/ două unităţi floppy/ XT, AT, 386 sau 486

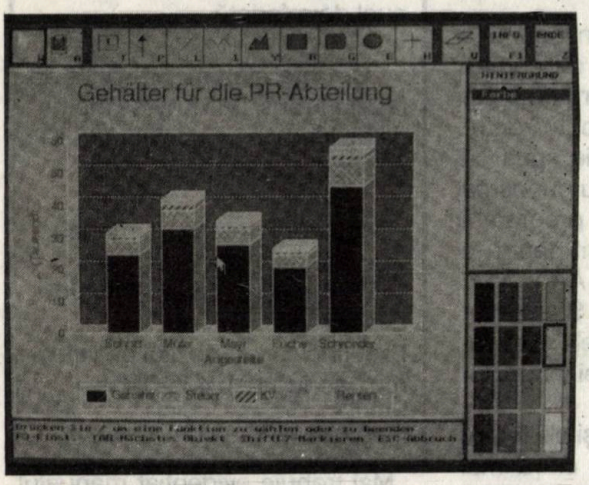
Operare: tastatură

Producător/ofertant: Computer Associates, Darmstadt

Preţ (inclusiv impozitul pe valoarea adăugată): cca. 2000 DM, reţea (3 posturi de lucru suplimentare) cca. 2230 DM.

astfel apărute. Ca urmare deservirea este anevoioasă şi greu de învăţat de către începători.

Quattro Pro 2.0



De-abia a apărut ce-a mai nouă versiune Quattro Pro a firmei Borland şi acest program agită deja lumea utilizatorilor. Şi această versiune a adus noi performanţe care vor pricinui, cu certitudine, multe nopţi albe concurenţei.

Pînă acum, Quattro Pro 2.0 este singurul program de calcul tabelar care uneşte avantajele unei suprafeţe utilizator grafice cu reprezentarea uzuală orientată caracter.

Pe de o parte el oferă tot ceea ce este reprezentativ, pentru Excel de ex., şi anume deservirea confortabilă cu mouse-ul, iar pe de altă parte necesarul de hard rămîne între limite rezonabile. Cu ajutorul setului de caractere ASCII, Borland reuşeşte să obţină performanţe surprinzătoare prin uşurinţa cu care poate crea ferestre, le poate deplasa şi le poate schimba dimensiunea cu ajutorul mouse-ului.

Cu toate că programul nu ştie gestiona foi de calcul tridimensionale, el ţine totuşi pasul cu Lotus 1-2-3 3.1 foarte bine, datorită funcţiilor sale de primă clasă cu care poate înlănţui foile de lucru. Starea suprafeţei de lucru poate fi memorată în mod asemănător ca la Apple Macintosh. În acest mod toate fişierele necesare într-o prelucrare pot fi încărcate dintr-o dată.

Întotdeauna Quattro Pro a avut de oferit performanţe ridicate în domeniul bazelor de date. În versiunea 2.0 acestea sînt şi mai evidente.

Pe lîngă interfaţa performantă spre SGBD-ul Paradox, al aceleiaşi case de soft, Borland oferă cu produsul său şi un modul SQL Frontend. Acest limbaj de interogare a bazelor de date, joacă un rol însemnat mai ales în cazul cuplării la calculatoare mari. Dar şi utilizatorul de PC-uri are de profitat de pe urma interfeţei cu bazele de date. Astfel pot fi accesate fişiere externe bazelor de date, începînd de la Paradox şi pînă la dBase cu toate versiunile lui, inclusiv dBase IV.

Limbajul macro foarte performant este cel care face ca programul de calcul tabelar Quattro Pro să devină baza ideală pentru dezvoltarea de aplicaţii. Cu ajutorul facilităţilor profesionale de eliminare a erorilor din macrouri, cum ar fi trasarea execuţiei şi depanorul, totul se rezolvă foarte rapid.

Quattro Pro devine un punct de referinţă şi datorită posibilităţilor sale de grafică. Există puţine programe special concepute pentru grafica comercială care să poată ţine pasul cu el.

Dacă domeniul de reprezentat este marcat, atunci programul se transformă dintr-un simplu program de calcul tabelar într-un program de prezentare grafică cu suprafaţă utilizator grafică. La marginea superioară a ecranului se află o "riglă iconografică" ce conţine uneltele de desen curente. Marginea din dreapta ecranului este ocupată de un meniu de alegere şi selecţie. Restul ecranului stă la dispoziţie pentru realizarea graficului.

Oferta de tipuri de grafice este completă şi nu există nici un detaliu care să nu poată fi stabilit de utilizator, indiferent că este vorba de culoarea coloanelor, de tipul ramelor sau de marcajul axelor X şi Y. Cu ajutorul uneltelor de desen cum ar fi cerc/elipsă, pătrat sau linie pot fi scoase în evidenţă anumite detalii, sau pot fi inserate anumite desene simple.

Dar Quattro Pro poate concura cu programele de grafică comercială nu numai datorită facilităţilor sale de grafică ci şi datorită opţiunii "Slideshow" care îl recomandă şi pentru realizarea de prezentări simple. Este singurul program de calcul tabelar care oferă şi o astfel de opţiune.

Quattro Pro 2.0

Furnitura de livrare: 6 dischete de 5,25", sau 3 dischete de 3,5", 3 manuale, şabloane de tastatură.

Cerinţe sistem: MS-DOS de la 2.11 în sus/ 512 Kbytes memorie de lucru/ XT, AT, 386 sau 486

Operare: tastatură, mouse

Producător/ofertant: Borland, München

Preţ (inclusiv impozitul pe valoarea adăugată): cca. 1500 DM

Starplaner 2.0

Cel mai ieftin dintre programele testate pînă acum este Starplaner 2.0. Casa de soft Star-Division din Lüneburg oferă prin acest program de calcul tabelar un clon solid al programului Lotus 1-2-3 2.01. Aceasta înseamnă că pe de o parte Starplaner 2.0 nu poate ţine pasul întru totul cu performanţele noilor versiuni Lotus, dar pe de altă parte

are de oferit şi ceva în plus faţă de original.

Diferenţele se observă încă de la pornire. La apăsarea tastei "CAPS-LOCK" nu apare meniul obişnuit Lotus ci un meniu modern de tip pull-down. Dacă faţă de model nu ar exista această corectură reuşită, am putea crede că ne aflăm în faţa programului Lotus 1-2-3, pentru că deservirea se face în mod identic, începînd de la tasta de apel şi pînă la structura meniurilor şi funcţii. Nu este de mirare că fişierele şi macrourile lui Starplaner 2.0 sînt compatibile cu cele ale lui Lotus 1-2-3.

Volumul de funcţii oferite de Starplaner nu poate ţine pasul cu concurenţa modernă. Nici posibilităţile de interconectare între fişiere nu există la Starplaner.

Pe de altă parte se poate pune întrebarea dacă toate aceste facilităţi sînt neapărat necesare. Unui utilizator obişnuit, sau unei întreprinderi mici îi sînt suficiente şi funcţiile oferite. Atît comercianţii cît şi oamenii de ştiinţă pot fi satisfăcuţi de funcţiile matematice, între acestea numărîndu-se chiar şi manipulări de matrici şi analiza regresiilor.

Starplaner 2.0 nu conţine multe dintre facilităţile oferite de concurenţă, cum ar fi: legătura cu baze de date externe şi macrouri. Pentru salvarea onoarei trebuie spus că nici Lotus 1-2-3 2.01 nu cunoştea aceste lucruri. Oricum programul Starplaner este suficient pentru o utilizare individuală. Adăugînd funcţiile de estimare, cu Starplaner se pot rezolva problemele generate de administrarea unei magazii.

Funcţiile grafice sînt şi ele foarte modeste. Lotus însuşi oferă mai mult numai datorită lui Allways. Sînt disponibile totuşi mai multe tipuri de grafice tip "coloană" şi "tort", dar ele nu ajung pentru scopuri de prezentare ci doar pentru analiza vizuală a unui şir de cifre.

Starplaner 2.0

Furnitura de livrare: două dischete 5,25", 360 Kbytes/ o dischetă 3,5", 720 Kbytes/ manual legat, cca. 400 pag./ manual de referinţă

Cerinţe sistem: MS-DOS de la 2.11 în sus/ 384 Kbytes memorie de lucru/ o unitate floppy/ XT, AT, 386 sau 486

Operare: tastatură, mouse

Producător: Star-Division, Lüneburg
Preţ (inclusiv impozitul pe valoarea adăugată): 400 DM

Mai trebuie evidenţiat manualul de prezentare, foarte bine realizat, şi nu trebuie pierdut din vedere preţul foarte scăzut la care este oferit produsul, preţ la care devine demn de atenţie.
(Computer Persönlich 26/90,
Richard Joerges)

Tabel comparativ

Nume	Excel 2.1	Lotus 1-2-3 2.2	Lotus 1-2-3 3.1	Starplaner 2.0	Supercalc5	Quattro Pro 2.0
Preţ aprox. (DM)	1700	1700	2050	400	2000	1500
Date tehnice						
Mărimea foii de calcul (linii/coloane)	16384x256	8192x256	8192x256x256	8192x256	9999x256	8192x256
Numărul fişierelor deschise simultan	func. de mem.	1	256	1	255	64
Foi de calcul tridimensionale	nu	nu	da	nu	da, limitat	nu
Necesită harddisk	da	nu	da	nu	nu	nu
Spaţiu necesar pe harddisk	1,5 MBytes	320 KBytes	4/5 MBytes	320 KBytes	1,2 MBytes	4 MBytes
WYSIWYG	da	cu Allways	da	nu	nu	nu
Operare cu mouse	da	nu	da	nu	nu	da
Prezentarea a mai mult de 24 de linii	da	da	da	da	da	nu
Ştie folosi coprocesorul	da	da	da	nu	da	da
Versiune pentru reţea	da	da	da	nu	da	da
Caracteristici de putere						
Tabela şi graficul pe aceiaşi foaie	nu	nu	da	nu	nu	da
Tabela şi graficul la o imprimare	nu	da, cu Allways	da	nu	da	da
Fonturi de diferite dimensiuni	da,4 simultan	da	da	nu	da	da
Listare Postscript	da	da	da	nu	da	da
Undo simplu	da	da	da	da	da	da
Undo multiplu, ierarhic	nu	nu	nu	nu	nu	nu
Culoarea suprafeţei modificabilă:	da,cu Windows	da	da	da	da	da
Copie de siguranţă la memorare	da	da	da	nu	da	da
Înregistrator de macrouri	da	da	da	nu	da	da

Haos şi fractali

Despre atractori "ciudaţi", cu dimensiuni fracţionare şi despre autoasemănare. Istoria haosului, ordinii şi a fractalilor.

Amintirea trecutului include în ea credinţa oamenilor de ştiinţă, de atunci, că totul în Univers este calculabil, că orice eveniment trecut, prezent sau viitor se poate prognoza. Lumea era privită ca un ceasornic uriaş în care fiecare realitate a prezentului îşi avea cauzele ei în trecut şi efectele sale în viitor. Şi totuşi... Încă în anul 1876, Maxwell atrăgea atenţia că nici un eveniment care se repetă nu poate fi identic în toate relaţiile dintre cauzele şi efectele sale. Cu alte cuvinte, nici un eveniment nu se poate repeta într-un mod absolut identic. Un vis se spulbera: cauzele asemănătoare nu mai generau efecte asemănătoare, Mici perturbaţii în condiţiile iniţiale ale unui sistem puteau avea implicaţii majore în evoluţia lui. Mici perturbaţii ale traiectoriei unui corp ceresc, datorate atracţiei altor corpuri aflate în apropiere, pot complica la nesfîrşit traiectoria lui viitoare.

În drum spre haos

Dar nu e nevoie să căutăm neobişnuitul în mişcarea stelelor, el poate fi găsit mult mai aproape. Să ne amintim de exemplu de pendulul matematic învăţat în anii de liceu. Cum poate fi descrisă mişcarea lui? O primă variantă ar fi aproximarea masei lui cu o masă punctiformă, independent de timp. Obţinem atunci arhicunoscuta sinusoidă. Dar această aproximare nu este suficientă. O altă idee ar fi să încercăm să descoperim care sînt mărimile ce variază în sistem, şi să găsim o relaţie între ele. De exemplu relaţia dintre unghiul făcut de pendul cu poziţia de echilibru şi viteza unghiulară din acel punct. (Amintim că viteza unghiulară este unghiul parcurs în unitatea de timp). Această relaţie o vom exprima cel mai uşor grafic, conform figurii 1.

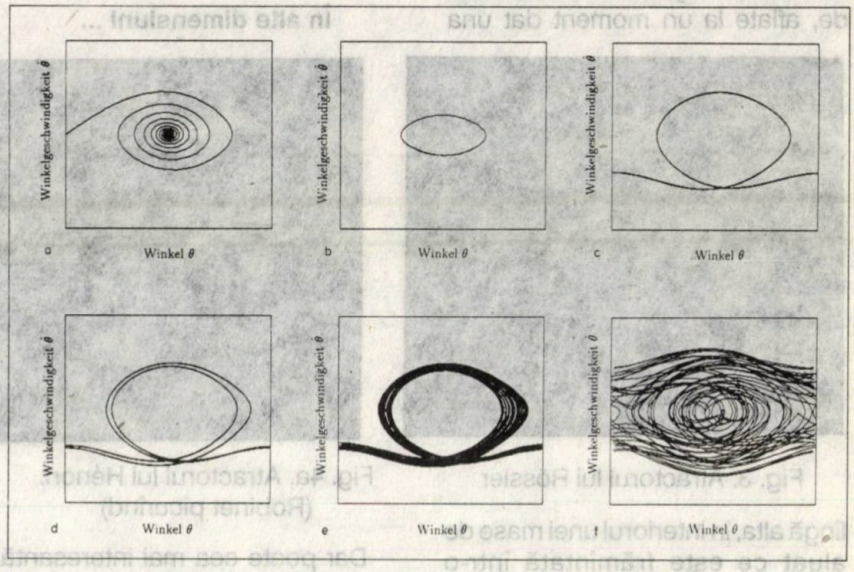


Fig. 1a-f: O pendulă in drumul spre haos

Figura 1a conţine cazul ideal în care masa este punctuală. Figura 1b arată ceea ce se obţine dacă mişcarea este menţinută constantă cu ajutorul unui motor extern. Dar realitatea este de fapt alta. Figura 1 punctele c-f ne arată ce se întîmplă în realitate, singura diferenţă între ele fiind unghiul iniţial din ce în ce mai mare la care este ridicat pendulul. Observăm apariţia unei mişcări haotice, deterministe şi dispariţia periodicităţii mişcării. Deterministe pentru că mişcarea rămîne totuşi calculabilă în ipoteza că am putea cunoaşte exact toţi parametrii sistemului. Şi mai observăm încă ceva: mişcarea se face ca şi cum ar exista undeva un punct de atracţie. Dar atracţia aceasta se comportă aparent haotic, de aceea vom numi acest fenomen drept apariţia unui atractor "ciudat". Aceşti atractori se întîlnesc des în natură. Să mai dăm cîteva exemple: Cel mai cunoscut dintre ei a fost descoperit în 1963 de către meteorologul Edward Lorenz de la MIT (Massachusetts Institute of Technology). Acesta încerca să dezvolte cu ajutorul calculatorului un model matematic de prognoză a vremii. Dar calculele erau mari şi computerul rula prea încet. Atunci lui Lorenz i-a venit ideea de a folosi rezultatele anterioare ca intrări pentru noile rulări. Surpriza a fost mare: s-a obţinut o cu totul altă prognoză. Unde era greşeala? În loc de 4 cifre după virgulă, Lorenz introdusese numai 3. O eroare de cîteva zecimi de miimi răsturnase complet prognoza lui. Exagerînd puţin: bătaia din aripă a unui fluture în China poate provoca cîteva săptămîni mai tîrziu un uragan în S.U.A.

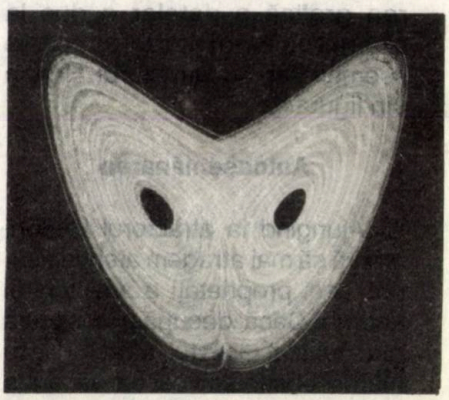


Fig. 2. Atractorul lui Lorenz

Exemplificare care a intrat în istorie ca "efectul de fluture". Imaginea acestui atractor se găseşte în figura 2.

Chimistul Otto Rössler a descoperit şi el un atractor "ciudat" încercînd să studieze un fapt mult mai banal: mişcarea a două stafide, aflate la un moment dat una

Fig. 3. Atractorul lui Rössler

lîngă alta, în interiorul unei mase de aluat ce este frămîntată într-o maşină specială. Ceea ce a obţinut este "atractorul Rössler" care arată ca în figura 3.

Alţi patru cercetători de la Universitatea Californiană din Santa Cruz au descoperit un atractor "ciudat" studiind un simplu robinet care picura. Şi anume au măsurat intervalele de timp dintre 4000 de picături succesive. Ce părere aveţi? Vor fi ele complet neregulate sau şi aici acţionează un atractor ciudat? După cum bănuiţi: şi în acest caz comportarea picăturilor de apă a fost haotică. Reprezentarea grafică a datelor a dus la obţinerea unui atractor foarte asemănător cu atractorul Henon din figura 4a.

Autoasemănarea

Ajungînd la atractorul Henon, merită să mai atragem atenţia asupra unei proprietăţi a acestuia şi anume: Dacă decupăm o bucată din graficul acestui atractor şi o mărim observăm că ea va arăta identic cu întregul (fig 4b). Această proprietate a părţii de a fi identică structural cu întregul, matematicienii au botezat-o autoasemănare. Un exemplu banal de autoasemănare este o dreaptă. Încercaţi să decupaţi o bucată de dreaptă şi să o măriţi. În noua imagine dreapta nu va fi nicidecum mai lată. Pentru că dreapta nu poate fi niciodată mai "lată" decît un punct...

În alte dimensiuni ...

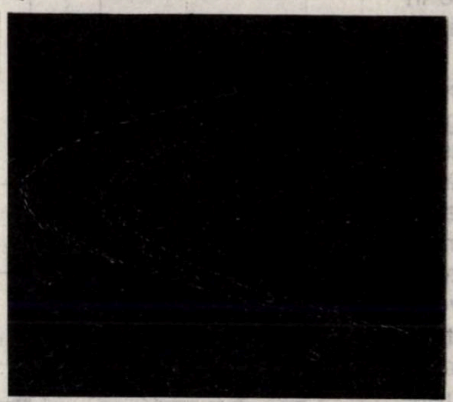


Fig. 4a. Atractorul lui Hénon. (Robinet picurînd)

Dar poate cea mai interesantă proprietate a atractorilor ciudaţi este alta: dimensiunea lor frontală nu este întreagă! Ce înseamnă asta? Euclid ne-a învăţat să percepem dimensiunile într-un anumit fel: o formă în spaţiu are trei dimensiuni, una în plan are două, pe o dreaptă există o singură dimensiune. Dar Euclid nu este singurul ochi prin care poate fi văzută lumea. Topologia ne aduce o definiţie recurentă a noţiunii de dimensiune a unei mulţimi. (Din punct de vedere topologic, corpurile sînt văzute ca mulţimi de puncte). În baza acesteia, o mulţime de puncte are dimensiunea n dacă poate fi separată printr-o mulţime de dimensiunea (n-1). Recursivitatea definiţiei se opreşte la mulţimile incorecte de puncte care se consideră a avea dimensiunea 0. Din punct de vedere topologic, orice curbă în spaţiu are dimensiunea 1. pentru că poate fi separată printr-un punct. O suprafaţă în spaţiu, tridimensională în geometria euclidiană, are dimensiunea 2 pentru că poate fi separată printr-o curbă de dimensiune 1. Dar să revenim la fractali. Calculul dimensiunii lor a fost fundamentat de Felix Hausdorff în 1919. Vom numi fractal orice mulţime de puncte în spaţiu care are dimensiunea neîntreagă (fracţionară). Să încercăm să înţelegem intuitiv acest lucru. Să presupunem că avem o curbă desenată pe o hîrtie milimetrică, şi că numărăm căsuţele hîrtiei prin care trece curba. Numărul lor va depinde categoric de lăţimea unei case-

Fig. 4b. O secţiune din linia atractorului lui Hénon

te de pe hîrtia milimetrică. S-ar putea chiar găsi o relaţie de proporţionalitate între N, numărul de casete atinse şi 1/e dacă notăm cu e lăţimea unei casete. O relaţie care ar avea o formă aproximativă N + C/e la n, unde C este o constantă. Ne-ar interesa puterea lui e, acel n mic, el ne va da în final dimensiunea fractală a curbei. Să ne gîndim la o dreaptă: acest n va fi 1, adică numărul de casete atinse va fi direct proporţional cu 1/e. Este un rezultat cunoscut. Dar să ne întoarcem la atractorii noştrii "ciudaţi". Ei umplu mult mai bine foaia de hîrtie. Dacă ar atinge toate punctele colii dimensiunea fractală ar fi 2 (N proporţional cu 1/e la puterea 2); dar nu ating toate punctele. Dimensiunea lor fractală se află undeva între 1 şi 2. Am putea concluziona că, intuitiv, dimensiunea fractală este gradul în care o mulţime de puncte umple spaţiul. (în exemplul nostru, coala de hîrtie).

Ar mai trebui făcute cîteva precizări. Unul din punctele de definiţie ale unui atractor "ciudat" este această dimensiune fractală neîntreagă. Atractorii ciudaţi apar

într-adevăr ca o expresie a comportării haotice a unui sistem. Dar nu trebuie să tragem de aici concluzia că orice sistem care se comportă haotic are un atractor "ciudat", şi nici că orice atractor ciudat determină comportarea haotică. Haosul este o proprietate dinamică a sistemelor. Acestea ar fi faptele. Dar pentru programatori, frumuseţea de-abia odată cu fractalii începe.

Fractali

Vom începe discuţia despre fractali cu un exemplu celebru, aşa numita "curbă a lui Kosch". Construcţia ei este uşor de observat în figura 5.

Pornim de la un segment, pe care îl împărţim în trei părţi egale. Partea din mijloc o vom elimina construind deasupra ei celelalte

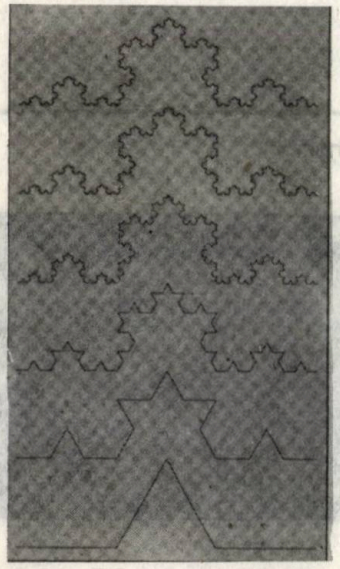


Fig. 5. Geneza unei curbe Kosch

două laturi ale unui triunghi echilateral. Noua figură obţinută are patru segmente. Fiecăruia dintre ele îi aplicăm acelaşi tratament şi obţinem o figură cu 16 segmente. Reluăm algoritmul la infinit. Ceea ce se obţine este un fractal. Dimensiunea lui este 1.26, deci este între o linie şi suprafaţă. O linie "plată" dacă acceptaţi exprimarea. Denumirea de fractal i-a fost dată acestei curbe de către Benoit Mandelbrot, pentru că dimensiunea fracţională a construcţiei rezultate este mai mare decît dimensiunea ei topologică (care este 1). O privire scurtă asupra construcţiei ne arată că aceasta este şi autoasemenea. Partea este o replică în miniatură a întregului. Metoda de construcţie poate fi uşor generalizată: ceea ce e important este că există o figură de bază şi un model de construcţie. Luaţi de exemplu drept bază tot un segment dar construiţi pe el un triunghi dreptunghic. Veţi obţine fractalul lui Minkowski.

Ce au în comun aceste construcţii este faptul că o singură transformare nu este îndeajuns. Transformările se aplică pe figurile obţinute succesiv la infinit. Propoziţia aceasta care descrie transformarea ce se aplică repetitiv poartă numele I.F.N. (Iterated Function System).

Frumosul în forme regulate

Exemplele de fractali ar putea continua încă multă vreme, dar to-

Fig. 6. Feriga după metoda lui Barnsley

tuşi vom stărui doar asupra unora mai spectaculoase. Iată unul: feriga matematică (fig. 6).

Ideea de bază a acestuia este folosirea unor transformări conforme. Se folosesc mai multe transformări. Aici apare însă şi o dificultate, e nevoie de un număr mare de iteraţii pentru a obţine o imagine clară, din cauză că structurile fine apar greu pe rezoluţia ecranului. Ar fi bine dacă s-ar putea să se construiască, pe cît posibil, mai întîi contururile, Şi această posibilitate a fost găsită prin aplicarea teoremei de convoluţie a lui Bernsley. Ideea este de a găsi un punct fix al transformării (care este lăsat pe loc de către transformare) şi să se pornească generarea din acest punct. Urmează apoi aplicarea iterativă a uneia sau alteia dintre transformări alese aleator. Figura devine "clară" mult mai repede. În figura 7 este exemplificat un program scris sub Turbo C, pentru obţinerea triunghiului lui Sierpinski. Singurul comentariu care trebuie făcut este că vectorul "greutate" a fost folosit pentru direcţionarea alegerii cu o probabilitate mai mare a uneia sau alteia dintre cele 3 transformări. Transformările se scriu matricial sub forma:

$$f(x) = \begin{matrix}(a_{11}\ a_{12}) \\ (a_{21}\ a_{22})\end{matrix} \begin{matrix}(x) \\ (y)\end{matrix} + \begin{matrix}(b_1) \\ (b_2)\end{matrix}$$

Julia, oh Julia!

Matematicianul Gaston Julia a folosit în 1918 următoarele egalităţi:

$$x' = x^2 - y^2 + a$$
$$y' = 2xy + b$$

scrise de el sub forma:

$$z' = z^2 + c$$

Aceste egalităţi neliniare pot fi folosite deasemenea pentru generarea unor fractali. Să fixăm un punct de pornire P(x,y) şi să lăsăm pe a şi b să fie oarecare fixaţi. Iterarea transformării ne poate duce la una dintre următoarele trei situaţii:

- 1.Numerele vor fi din ce în ce mai mici.
- 2.Numerele vor fi din ce în ce mai mari şi tind la infinit.
- 3.Numerele se menţin pe traiectorii stabile şi mărginite.

```
#include <graphics.h>
#include <math.h>
#include <stdlib.h>

double gew[] = { 0.25, 0.5, 0.75 };
double gewsum;
double a11[] = { 0.5, 0.5, 0.5 };
double a12[] = { 0.0, 0.0, 0.0 };
double a21[] = { 0.0, 0.0, 0.0 };
double a22[] = { 0.5, 0.5, 0.5 };
double b1[] = { 0.0, 0.5, 0.25 };
double b2[] = { 0.0, 0.0, 0.43301 };
double x, y;
int color;

void nextpoint ( double r );

main()
{   int nx, ny;
    double xmax, ymax, xmin, ymin;
    int grdrv = DETECT, grmode;
    int i;
    initgraph( &grdrv, &grmode, "" );
    ny = getmaxy();
    nx = getmaxx();
    xmax = 1.0;
    ymax = 1.0;
    xmin = 0.0;
    ymin = 0.0;
    for ( gewsum = 0 , i = 0 ; i < sizeof(gew) / sizeof(double) ; i + + )
        gewsum + = gew[i];
    while ( 1 )
    {
        double r;
        r = ( ( double )rand() ) / RAND_MAX;
        nextpoint(r);
        putpixel( nx * ( x + xmin ) / ( xmax + xmin ) ,
            ny * ( y + ymin ) / ( ymax + ymin ) , WHITE );
        if ( kbhit() )
        {
            getch();
            break;
        }
    }
    closegraph();
    return(0);
}

void nextpoint( double r )
{
    int index;
    double sum;
    index = 0;
    sum = gew[0];
    while ( sum < r * gewsum )
    {
        index + +;
        sum + = gew[index];
    }
    color = index + 2 ;
    sum = a11[index] * ( double ) x + a12[index] * ( double ) y +
            b1[index];
    y = a21[index] * ( double ) x + a22[index] * ( double ) y +
            b2[index];
    x = sum;
}
```

Fig. 7. Program C pt. desenarea triunghiului lui Sierpinski

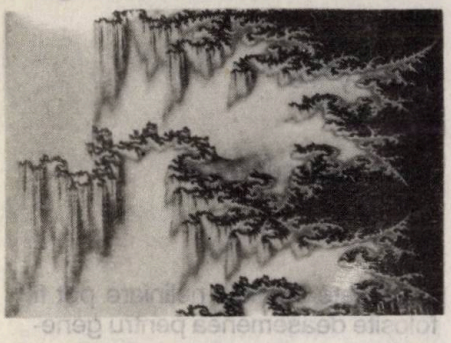


Fig. 8. Fractalul San Marco ca exemplu de mulţime Julia

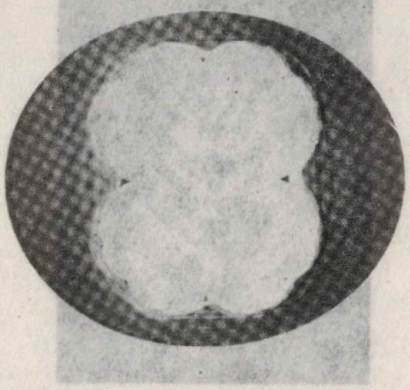


Fig. 9. Un alt exemplu de mulţime Julia

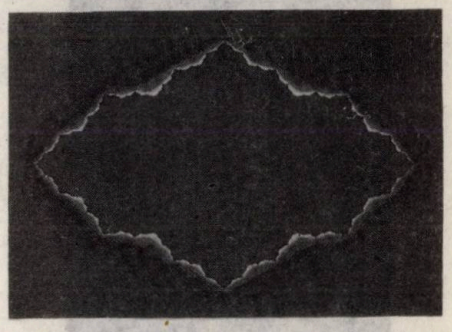


Fig. 10. Un fractal obţinut după ideea lui Mandelbrot

Mulţimea punctelor P(x,y) care cad peste punctul 3 va fi numită mulţimea Julia şi o vom nota cu J(a,b). J(a,b) este şi ea un fractal şi ne arată o imagine încîntătoare. (figurile 8,9)

Iteraţiile sînt şi în acest caz foarte multe. Mai tîrziu Mandelbrot, despre care am mai vorbit, a venit cu ideea ca a şi b să fie utilizate ca punct de pornire. Să admirăm încă o dată, în final o ultimă imagine a unui fractal obţinut după ideea lui Mandelbrot în figura 10.

(PC + Technik, 1/91, Jobst Kehrhahn)

Unul dintre corespondenţii revistei - Ioan Tiberiu Socaciu, student în Cluj - a venit în întîmpinarea acestui articol trimiţîndu-ne un program scris în Turbo Pascal, pentru generarea fractalului lui Kosch. În cele ce urmează vă prezentăm programul său şi vă mai prezentăm un program de desenare fractali pe monitoare VGA.

```
{$n-}
uses crt, graph3, graph;

type punct = record x,y: real end;
var a,b: punct;
const max = 3;
type sir = array[1..max] of punct;
var n:integer;

procedure puncte(a,b:punct; var t:sir; var n:integer);
var aa,xr,yr,l,sc:real;
begin
n: = 3;
t[1].x: = (2*a.x + b.x)/3;
t[1].y: = (2*a.y + b.y)/3;
t[3].x: = (2*b.x + a.x)/3;
t[3].y: = (2*b.y + a.y)/3;
aa: = sqrt(sqr(a.x-b.x) + sqr(a.y-b.y));
l: = aa/3;
sc: = l/aa;
xr: = (a.x + b.x)/2;
yr: = (a.y + b.y)/2;
t[2].x: = xr + sc*(b.y-a.y);
t[2].y: = yr-sc*(b.x-a.x);
end;

procedure plt(a:punct);
begin
with a do plot(trunc(x), trunc(y), 1)
end;

function dist(a,b:punct):boolean;
const eps = 1;
begin
dist: = (eps(sqrt(sqr(a.x-b.x) + sqr(a.y-b.y))))
end;

procedure fractal(a,b:punct);
var i:integer;
var t:sir;
begin
if dist(a,b) then

    puncte(a,b,t,n);
    fractal(a,t[1]);
    for i: = 1 to n-1 do begin fractal(t[i],t[i + 1]);plt(t[i]) end;
    plt(t[n]);
    fractal(t[n],b);
    plt(b)
    end;
end;

begin
read(a.x, a.y, b.x, b.y);
graphmode;
plt(a);
fractal(a,b);
repeat until keypressed;
end.
```

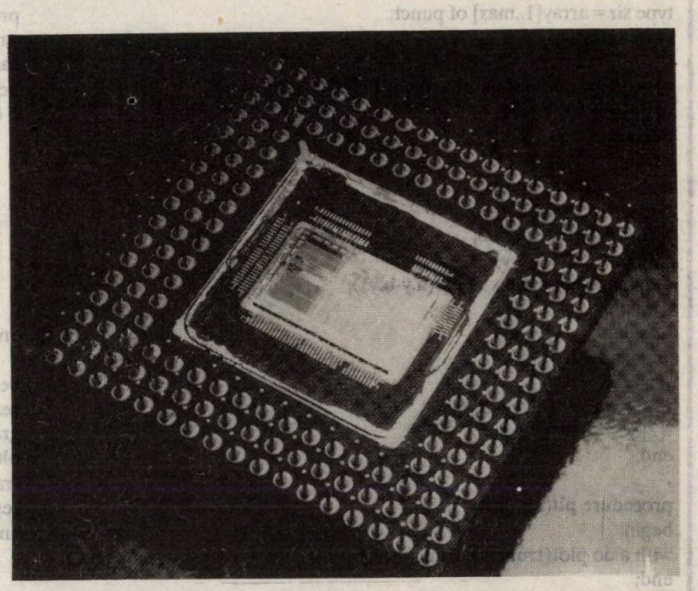
Fractali pe VGA

Cu ajutorul programului VGA.PAS, puteţi obţine pe un monitor VGA, grafice frumoase cu mijloace simple. Programul are nevoie de mult timp pentru a-şi face treaba, dar rezultatul merită să fie văzut. La lansarea în execuţie trebuie să aveţi grijă ca fişierul EGAVGA.BGI să se afle în acelaşi director cu programul. Dacă acest lucru nu se întîmplă programul este întrerupt cu mesajul de eroare: "bgi-error: Graphics not initialized". Graficul obţinut poate fi modificat foarte uşor, modificînd condiţia pe care trebuie s-o îndeplinească punctul ce va fi afişat. Pentru aceasta este suficient să se înlocuiască în rutina principală apelul la funcţia Def1 cu un apel la una din funcţiile: Def2 sau Def3. Programul poate fi rulat şi pe alte plăci grafice, modificînd valorile xmax, ymax şi testul de driver.

(PC + Technik 1/91,
Michael Podlejska, Franz Gutsis)

```
{$B-,R-,S-,I-}

PROGRAM  vga;

uses crt,graph;
const    xmax = 640;
         ymax = 480;
var x,y,x_end,y_end,f,xp1,yp1,xp2,yp2: word;
    driver, modus: integer;
    a,b,co,si,t: real;

function ra (arg: real): word;
begin
    ra: = round (abs (arg));
end;

procedure def1;
begin
    t: = x*(x + y)/20;
    co: = cos(t); si: = sin(t);
    a: = co + t*si; b: = si-t*co;
    if a > b then f: = ra(y*y/320400*t/50) mod 7
        else f: = 6 + ra(x/640*t/100) mod 7;
end;

procedure def2;
begin
    a: = cos(0.02*y/(2-sin(x*y/2400)));
    f: = ra(15*a);
end;

procedure def3;
begin
    a: = (cos(0.087*y*y/(161-0.5*x))*
            sin(0.005*x*x/(101-0.41*y)));
    f: = abs (round(15*a));
end;

begin
    driver: = detect;
    DetectGraph (driver, modus);
    if driver < > vga then
    begin
        writeln ('Placa grafic{ nu este VGA !');
        halt;
    end;
    modus: = vgahi;
    InitGraph (driver, modus, '');
    SetBkColor (black);
    x_end: = (xmax div 2) - 1;
    y_end: = (ymax div 2) - 1;
    y: = 0;
    while y < = y_end do
    begin
        if KeyPressed then halt;
        x: = 0;
        while x < = x_end do
        begin
            def1;
            if f < > 0 then
            begin
                xp1: = x + x_end + 1;
                yp1: = y + y_end + 1;
                xp2: = x_end-x;
                yp2: = y_end-y;
                PutPixel (xp1,yp1,f);
                PutPixel (xp1,yp2,f);
                PutPixel (xp2,yp1,f);
                PutPixel (xp2,yp2,f);
            end;
            inc (x);
        end;
        inc (y);
    end;
    sound (1000);
    delay(1000);
    nosound;
    repeat until KeyPressed;
    CloseGraph;
end.
```

Un Cray pe 1,5 centimetri pătraţi - procesorul Intel i860



i860 într-o carcasă ceramică cu 168 de pini

Doar cu două decenii în urmă un calculator ultrarapid costa pînă la 50 de milioane de mărci. În afară de Institutele de Fizică Atomică, aceste calculatoare mai erau folosite doar în construcţiile de avioane pentru rezolvarea, de exemplu, a problemelor de aerodinamică cum ar fi configurarea suprafeţelor portante. Cu acestea, posibilităţile de aplicabilitate ale acestor calculatoare speciale erau în mare epuizate.

În anii '80 au apărut şi alte posibilităţi de utilizare pentru aceste calculatoare ultrarapide. Domeniile graficii, CAD, simularea circuitelor de comutare şi proiectarea cablajelor (autorouting) necesitau performanţe de calcul corespunzătoare din partea hard-ului. Ca urmare s-au dezvoltat, ca staţii de lucru, calculatoare care rezolvă extrem de rapid anumite clase de probleme. Astfel de calculatoare se pot obţine deja începînd de pe la 50.000 DM.

Cu circuitul 80860 Intel a realizat, la începutul lui 1989, un cip, care a produs mare nelinişte pe piaţa de staţii de lucru (workstations). Facilităţi rezervate pînă acum doar supercalculatoarelor sînt acum reunite într-un singur cip. Faptul că acest lucru este rezultatul unui timp de proiectare îndelungat şi al unui mare efort financiar rezultă din cîteva cifre.

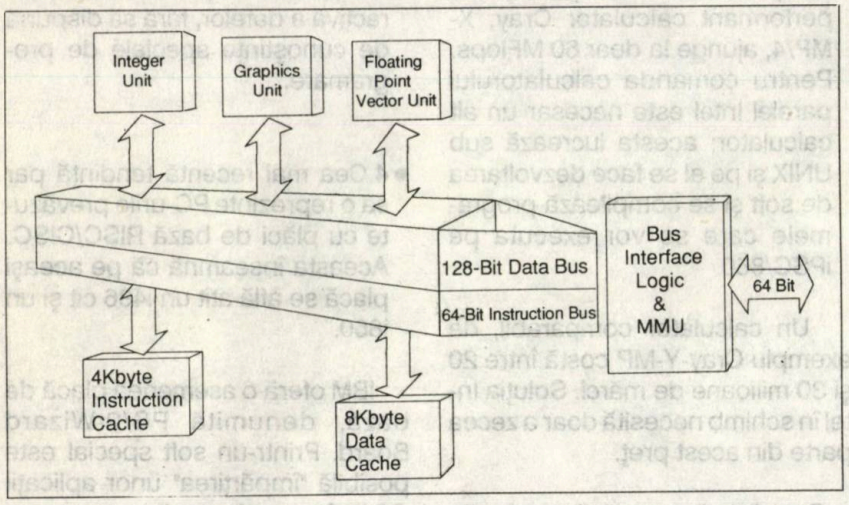
Proiectarea "supercipului" a necesitat 5 ani. S-au investit, în acest timp, circa un miliard de mărci. O treime din această sumă a fost "înghiţită" doar de fabrica în care se va produce acest cip. Căci o dată cu creşterea gradului de integrare a unui cip cresc şi necesităţile de echipamente pentru producerea lui. În cazul lui i860 s-au integrat, pentru prima dată într-o singură capsulă, un milion de tranzistoare. Pentru aceasta au fost necesare structuri de mărimea unui micron, adică a unei milionimi de milimetru. Paralel cu dezvoltarea acestei componente integrate la scară înaltă, s-a lucrat şi la un procedeu de fabricaţie care a făcut posibilă producerea acestui cip. De altfel, acelaşi procedeu a permis mai apoi şi producerea cip-ului 80486 al lui INTEL. În timp ce acesta din urmă este conceput pentru implementare în calculatoarele personale, 80860 a fost conceput ca o unitate centrală RISC (cu set redus de instrucţiuni) pentru staţii de lucru şi minicalculatoare pentru domeniile tehnico-ştiinţifice. RISC înseamnă "Reduced Instruction Set Computer" , un calculator ce lucrează cu un set redus de instrucţiuni. Această arhitectură nu este nouă, dar cunoaşte acum un enorm avînt. Abia dacă există producător de calculatoare renumit care să nu ofere sisteme bazate pe tehnologia RISC. Şi pentru PC-uri există între timp o serie de plachete coprocesor care transformă practic chiar şi AT-urile în staţii de lucru. Cu puţin timp în urmă Toshiba a ieşit pe piaţă cu primul Laptop dotat cu un cip de tip RISC. Această unitate centrală cu ceas de 20 MHz ajunge, împreună cu un coprocesor al firmei Weitek, la performanţe de 13,2 Mips şi 1,5 MFlops. Calculatorul se poate cumpăra (deocamdată) doar în S.U.A. şi Japonia la preţul de cca. 13.000$.

Fără îndoială că 80860 al lui INTEL este momentan cel mai performant reprezentant al unităţilor centrale RISC. Arhitectura acestuia corespunde în mare parte necesităţilor sistemului de operare UNIX. Spaţiul liniar de adresare de 4 GByte este împărţit în aşa numite pagini de 4 KByte. În plus unitatea de management a memoriei cunoaşte două regimuri de privilegiere: supervizor şi utilizator.

Pe cip sînt integrate în total trei procesoare: o unitate aritmetică, o unitate de virgulă flotantă şi o unitate grafică. La acestea se adaugă două memorii cache, (separat una pentru date şi una pentru instrucţiuni), precum şi unitatea de management a memoriei.

Magistrale (bus-uri)

Transferul intern extrem de rapid este asigurat de un bus de date de 128 biţi şi un bus de instrucţiuni de 64 biţi. (Spre comparaţie: magistrala internă de date a lui 80486 este de 32 biţi). La i860 se pot prelua simultan două instrucţiuni pe bus-ul de instrucţiuni, (de 64 de biţi), din cache-ul de instrucţiuni. pe un singur tact. În acest mod se poate ajunge la o viteză maximală de 960 Mbyte/sec. Dacă instrucţiunea nu se află în cache va fi preluată prin bus-ul extern. Microprocesorul i860 are o interfaţă sincronă la care bus-ul de date şi de adrese sînt separate. La 8086 octeţii "low" (ponderile mici) de adrese şi datele se aflau pe un bus



Schema bloc a procesorului RISC i860

comun de adrese. Unitatea centrală RISC lucrează cu un bus bidirecţional de date de 64 biţi şi un bus de adrese de 32 biţi. Cu acesta se poate adresa un spaţiu de adrese de 4 GBytes. Printr-un "bus-pipelining" în două trepte pot lucra simultan trei cicluri de instrucţiune.

Memoriile cache.

Unitatea centrală i860 are intern trei memorii cache: una pentru date, una pentru instrucţiuni şi una pentru unitatea de management a memoriei. Aceasta este folosită la translatarea rapidă a adreselor virtuale în adresele fizice corespunzătoare.

Raportul de regăsire la cache-ul de instrucţiuni de 4 kbytes este de 90 la sută. Aceasta înseamnă că în nouă din zece cazuri instrucţiunea se află în cache şi nu trebuie accesată memoria centrală pentru a o regăsi. Cache-ul de instrucţiuni şi de date folosesc procedura "Two Way Set Associativ". În pricipiu prin acest procedeu memoria centrală este împărţită în aşa numite pagini, care au mărimea unei jumătăţi de cache.

Unitatea aritmetică.

Unitatea aritmetică integrată ca RISC-CPU execută majoritatea instrucţiunilor şi răspunde de derularea programelor. Ea ajunge la 33 VAX Mips şi la peste 85.000 Dhrystones/sec, la o frecvenţă de tact de 40 MHz. Se pot încărca atît instrucţiuni relative la întregi cît şi la numere în virgulă flotantă, în total fiind disponibili 32 de registre de cîte 32 biţi.

Acest număr relativ mare de registre este tipic pentru arhitecturile RISC. În comparaţie cu acestea 80486 are 8 registre. Prin numărul mare de registre se ajunge ca aproape toate instrucţiunile, (aritmetice, logice, de comparare) să se deruleze între registre. Este cunoscut faptul că registrele interne ale unui procesor sînt elementele de memorie cele mai rapide. Datele din ele pot fi accesate fără cicluri de bus, ceea ce îmbunătăţeşte timpul de execuţie al instrucţiunii. Doar instrucţiunea de tip load/store mai lucrează cu accese la memorie şi deci necesită un timp suplimentar.

Unitatea de virgulă flotantă.

Unitatea de virgulă flotantă este coprocesorul lui i860 şi realizează o performanţă de 80 MFlops. Setul propriu de registe ale acesteia constă din 32 registe de 32 biţi. Calculele sînt realizate paralel de către circuite de înmulţire şi sumatoare. Prin aceasta se obţin două rezultate pe tact.

Un alt mod de prelucrare paralelă este aşa numitul "Dual Instruction Mode". Prin acesta unitatea aritmetică şi cea de virgulă flotantă lucrează simultan. Multe aplicaţii de virgulă flotantă permit accelerarea prin derularea în mod "pipelined" a procesului şi furnizează un rezultat pe tact. În acest mod se pot începe noi cicluri de instrucţiune cu toate că ciclurile instrucţiune lansate deja nu sînt încă terminate.

Unitatea de grafică 3D

Unitatea de grafică este optimizată pentru reprezentări tridimensionale şi umbriri. Pentru aceasta s-a prevăzut un set de cinci instrucţiuni grafice elementare. Prin utilizarea pentru calcule a unităţii de virgulă flotantă din cip, unitatea grafică poate lucra paralel cu unitatea aritmetică. Ea prelucrează simultan 8 pixeli (puncte) de 8 biţi, patru de 16 biţi sau 2 de 32 de biţi. Suprafaţa unui obiect masiv v-a fi reprezentată prin poligoane care corespund suprafeţelor obiectului.

Se pot reprezenta pînă la 40.000 de asemenea poligoane pe secundă. Eliminarea liniilor acoperite şi suprafeţelor acoperite are loc cu algoritmul Z-buffer.

Unitatea de management a memoriei.

La translatarea adreselor virtuale în adrese fizice se asigură o compatibilitate totală cu procesoarele 386 şi 486.

Prin unitatea de management a memoriei din cip nici timpul de execuţie al instrucţiunilor şi nici ciclul de bus datorită procesului de translatare a adreselor nu se va prelungi.

La microprocesoarele cu management de memorie externă unitatea centrală trebuie să furnizeze întîi adresele pe busul de adrese înainte de a se putea calcula valoarea adresei fizice. Pentru acest lucru este nevoie de cel puţin un tact. Translatarea adreselor la 860 conţine toate funcţiile de bază necesare pentru un sistem de memorie virtuală orientat pe pagini. Ca şi "page frame" (cadru pagină) se adresează un bloc contiguu de 4 kbyte.

Aplicaţii pentru i860

Abia dacă trece o săptămînă, în care să nu apară o aplicaţie nouă pentru cip-ul Intel i860. Existenţa unei oferte de soft din ce în ce mai mare are ca urmare în mod natural răspîndirea rapidă a cip-ului.

Aplicaţiile se pot împărţi în trei domenii principale:

- 1.Plachete cu un procesor 860 care pot fi integrate în calculatoarele existente, ca şi plachete coprocesor pentru ca să preia părţile de program cu calcule laborioase sau să faciliteze aplicaţiile grafice.

* De exemplu placheta Genisys-SPC 860 poate fi pusă într-un calculator compatibil AT şi va da posibilitatea unei măriri considerabile a performanţelor calculatorului la aplicaţiile grafice sau necesitînd calcule laborioase. Alte asemenea plachete sînt oferite de firmele Kontron sau Spea.

- 2.Un supercalculator al lui Intel, care utilizează mai multe unităţi centrale 860 în paralel, este iPSC/860. Într-o carcasă se pot dispune pînă la 64 plachete-procesor care lucrează independent.

Două asemenea "turnuri de calcul" legate împreună oferă 128 de procesoare RISC care lucrează în paralel unul lîngă altul. Astfel se ajunge la performanţe de aproximativ 5 GFlops. Cel mai performant calculator Cray, X-MP/4, ajunge la doar 60 MFlops. Pentru comanda calculatorului paralel Intel este necesar un alt calculator; acesta lucrează sub UNIX şi pe el se face dezvoltarea de soft şi se compilează programele care se vor executa pe iPSC/860.

Un calculator comparabil, de exemplu Cray-Y-MP costă între 20 şi 30 milioane de mărci. Soluţia Intel în schimb necesită doar a zecea parte din acest preţ.

Beneficiarii potenţiali provin din domeniile construcţiei de avioane, tehnicii spaţiale sau universităţi. Chiar şi la München se lucrează pe un calculator iPSC la dezvoltări de soft pentru calculatoare paralele.

- 3.Staţiile de lucru construite în jurul unor unităţi centrale RISC le elimină din ce în ce mai mult pe cele construite pe bază de CISC (Complex Instruction Set Computer).

Acestea din urmă utilizează un set cuprinzător de instrucţiuni, cu instrucţiuni complexe care trebuie apoi prelucrate de către un microprogram. În timp ce arhitectura RISC caută ca pe cît posibil să execute o instrucţie pe tact, la un calculator CISC sînt necesare pentru acesta mai multe tacte.

Recent, firma Stardent a prezentat o staţie de lucru avînd 4 unităţi centrale RISC. Între acestea se află pe lingă un R3000 pentru operaţii aritmetice al firmei Mips, un R3010 (tot Mips) pentru unitatea de virgulă flotantă şi două i860 care "răspund' de aplicaţii grafice.

Datorită acestora sistemul - de mărimea unui calculator "desktop" devine un supercalculator cu grafică de mare performanţă, la preţul unei staţii de lucru. Se adaugă la toate acestea un sistem de vizualizare care permite utilizatorului reprezentarea interactivă a datelor, fără să dispună de cunoştinţe speciale de programare.

- 4.Cea mai recentă tendinţă par să o reprezinte PC-urile prevăzute cu plăci de bază RISC/CISC. Aceasta înseamnă că pe aceaşi placă se află atît un i486 cît şi un i860.

IBM oferă o asemenea placă de bază, denumită PS/2-Wizard Board. Printr-un soft special este posibilă "împărţirea" unor aplicaţii OS/2 între cele două procesoare. Procesorul Intel i860 preia partea de calcul laborios şi de grafică. În acest fel se pot obţine, la aplicaţii tehnico-ştiinţifice sub OS/2, performanţe care pînă acum le erau accesibile doar posesorilor de staţii de lucru din clasa superioară.

Primul PC care a aşezat procesorul 486 şi procesorul Intel RISC pe aceaşi plachetă provine din Olanda. În Germania firma este cunoscută sub numele de Borsu Computer.

(PC plus Technik 12/90, Hermann Staubinger)

State of the art

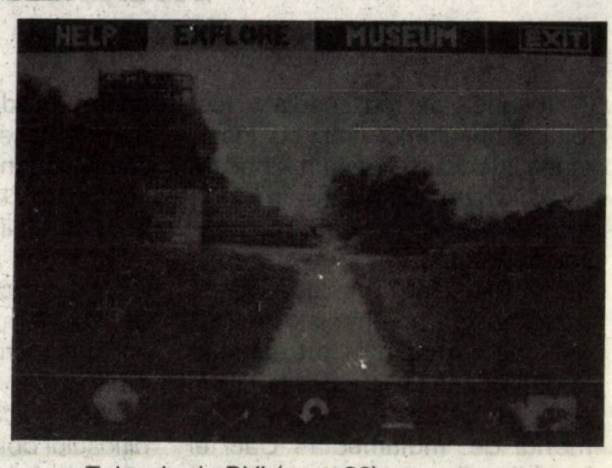
Sub titlul **"State of the art"** vă prezentăm 5 concepte hard şi soft care, în opinia noastră, vor juca în viitor un rol decisiv în lumea calculatoarelor.

Conceptele descrise aici au fost, în 1990, sau transpuse pentru prima dată într-un produs, sau au reuşit să se impună în cursul acestui an.

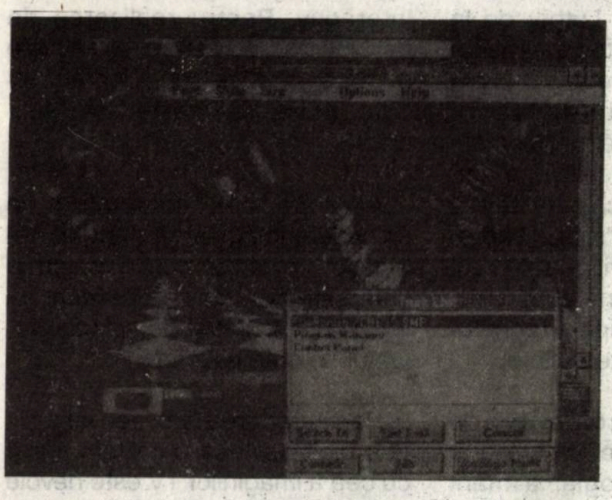
Pe baza criteriilor noastre de selecţie am păstrat în final 5 concepte care fie vor accentua în viitor o tendinţă existentă, fie vor putea chiar constitui bazele unei tendinţe noi. Caracteristica determinantă a trebuit să fie în fiecare caz importanţa pentru viitor. Pe viitor, aceste concepte fie se vor folosi de tehnologii noi, fie le vor folosi, în forme modificate, pe cele existente, în domenii noi şi aplicaţii neprevăzute iniţial.

Ordinea prezentării pe următoarele pagini a conceptelor şi tehnologiilor nu trebuie considerată a fi o ordine ierarhică. Fiecăruia i s-a alocat acelaşi loc. Înainte de prezentarea lor, sînt date cîteva informaţii de culise despre istoria genezei lor şi despre autori. Pentru a face mai clar de ce am ales exact aceste 5 concepte şi nu altele, explicăm tehnica sau concepţia şi enumerăm posibilităţile generate prin noul concept pentru dezvoltări viitoare în lumea calculatoarelor. Vom însoţi conceptele "noastre" pe parcursul anului 1991, vom prezenta produse noi şi vă vom relata la sfîrşitul anului 1991 în ce măsură selecţia noastră s-a dovedit justă.

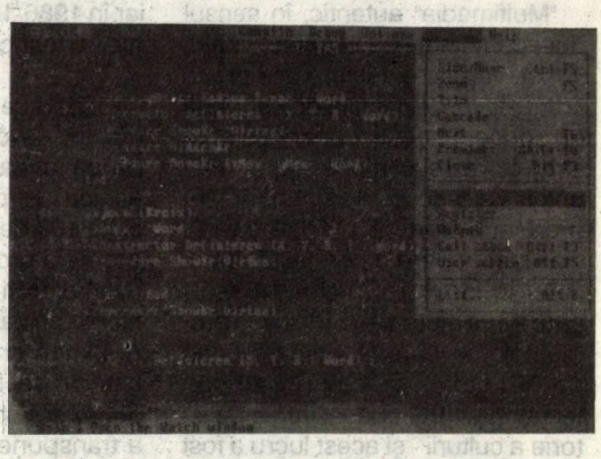
(PC+Technik, 1/91)



Tehnologia DVI (pag. 30)

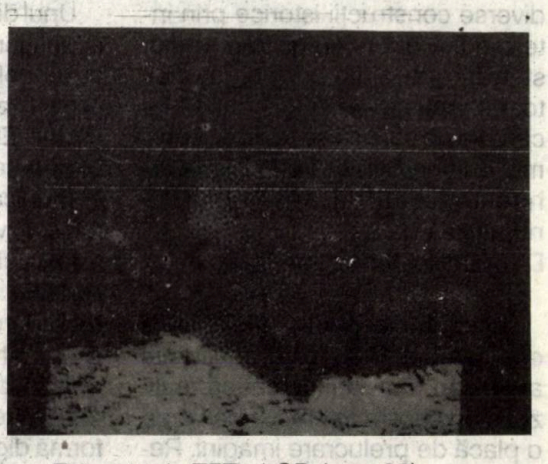


Windows 3.0 (pag.34)



Tehnologia Flash (pag.36)

Programarea orientată obiect (pag.38)



Tehnologia TFT - LCD (pag.32)

Multimedia Show

Noţiunea de multimedia a fost folosită în ultimul timp cu foarte multe înţelesuri, uneori mult diferite. Cînd un elaborator de soft reuşea de exemplu să integreze imagini într-o bază de date, vorbea de "multimedia". Şi în legătură cu videorecordere controlate de PC-uri - de exemplu la prezentarea unor produse pe la expoziţii, tîrguri etc. - se foloseşte cu plăcere termenul de "multimedia". Căci el înnobilează - sună bine - şi lasă mult loc pentru interpretări.

"Multimedia" autentic, în sensul unei combinaţii de imagine, video, sunet, vorbire şi texte, toate prelucrate digital pe calculator, este însă un concept nou. În concepţia noastră, el va revoluţiona lumea PC-urilor. Căci se poate imagina fără prea mult efort că într-o fereastră, via video, apare o persoană atractivă - în locul unui help-text sec - care să vă ajute să vă rezolvaţi problemele pe care le întîlniţi lucrînd pe PC. La fel, de ce nu v-aţi perfecţiona pe PC cunoştiinţele de istorie a culturii - şi acest lucru a fost deja demonstrat de Intel într-o manieră impresionantă - parcurgînd diverse construcţii istorice prin intermediul unor înregistrări video, stabilind cu ajutorul unui joystick tot ce vreţi să vedeţi şi care să fie cărările pe care să se realizeze drumul dumneavoastră prin istoria arhitecturii. Întreg ansamblul se numeşte
DVI - Digital Video Interface.

Începuturile pentru multimedia datează din 1983, cînd concernul american RCA a încercat să realizeze un sistem interactiv bazat pe o placă de prelucrare imagini. Rezultatele însă nu au fost satisfăcătoare. Larry Ryan a fost apoi primul care a propus realizarea unui sistem digital complet, care să integreze film, video, vorbire, muzică, imagini, grafică şi text. Împreună cu Art Kaimann, specialist în digitalizare, pe atunci director pentru "produse digitale" la David Sarnoff Research Center din Princeton, a lansat constituirea unei echipe de cercetare. La începutul lui 1984, lor li s-a alăturat specialistul în medii Arch C. Luther de la RCA Broadcast Systems Division din Candem, New Jersey. Pentru echipa de proiectanţi au urmat vremuri grele. Pe lîngă problemele tehnice ridicate de realizarea ambiţiosului obiectiv, echipa a trebuit să se confrunte şi cu "rotaţia cadrelor": patronul se schimba mereu. RCA a vîndut la General Electric, iar în 1986/1987 echipa şi tehnologia au fost cumpărate de Intel.

Deoarece Intel şi-a dat seama de efectele revoluţionare pe care le-ar putea avea tehnologia, a mărit echipa de proiectare de la 40 la circa 100 de colaboratori, pentru a accelera progresele. Cei mai mulţi colaboratori din perioada de pionierat, ca Larry Ryan, Art Kaimann, Al Simon şi Al Korenjak, au adus tehnologia DVI pînă în stadiul ei actual şi sînt şi acum la Intel pentru a transpune în fapte visul despre un calculator multimedia.

Unul din primele sisteme apte de a funcţiona şi a fi vîndute, realizate cu tehnologia DVI a Intel-ului, a fost prezentat publicului la Orgatec (Köln). El permite utilizarea simultană a celor mai diverse medii de comunicare - video, film, grafică, muzică, vorbire, zgomote, imagine şi text - şi aceasta fără a recurge la aparate suplimentare ca videorecordere sau casetofoane. Întregul show multimedia se desfăşoară pe PC; corespunzător, datele trebuie să fie disponibile în formă digitalizată, pentru a putea fi prelucrate pe calculator.

Hard-ul DVI constă din două plăci tip AT. Placa de preluare (capture board) face posibilă digitalizarea şi preluarea datelor video sau audio analogice. Informaţiile pot fi culese direct, cu un microfon sau o cameră de luat vederi, dar şi de pe un videorecorder. Placa de redare (delivery board), al doilea adaptor AT, permite redarea de video, audio şi imagine într-un mediu interactiv. Pentru realizarea de aplicaţii DVI, utilizatorul are nevoie de ambele plăci; numai pentru redare, placa de redare este suficientă.

Aproape toate informaţiile, fie ele video, imagine sau audio, trebuiesc prelucrate de PC în formă comprimată. Căci - după cum vă imaginaţi - la astfel de aplicaţii volumul de date este imens.

Pentru redarea unor secvenţe video digitalizate la o calitate egală cu cea a imaginilor TV este nevoie de circa 720 kbytes de memorie pentru o imagine. Şi deoarece sînt necesare, într-o secundă, 30 sau 25 de imagini (în sistemul NTSC respectiv în sistemul PAL) pentru a realiza mişcare, trebuiesc prelucrate peste 22 Mbytes de date într-o secundă. Necesarul imens de memorie face ca o prelucrare pe PC să aibă sens numai dacă sînt disponibile procedee eficiente de comprimare.

Informaţiile audio şi imaginile se pot comprima cu algoritmul de compresie specific DVI direct în PC, deoarece pentru acest tip de date algoritmii nu consumă foarte multă putere de calcul. Pentru date video, a căror compresie necesită un efort de calcul mult mai mare, sînt disponibile două variante: pe de o parte, o comprimare în PC, pe de altă parte, o comprimare realizată la Intel, pe un calculator paralel. Prima permite o rezoluţie de 128 * 120 puncte, a doua de 256 * 240 puncte. Pentru derularea aplicaţiei multimedia, şi în acest caz un PC uzual este suficient.

După compresia datelor, din cei 22 Mbytes/secundă video rămîn 150 kbytes (factorul de compresie

160:1 !!). Zece secunde de video necesită în consecinţă 1,5 Mbytes. Dacă vreţi să urmăriţi un videofilm de o oră, PC-ul d-voastră trebuie să dispună de cel puţin 600 Mbytes de memorie.

Cu memorii de masă moderne ca CD-ROM, WORM sau discuri MO, problemele de memorie vor putea fi rezolvate. Acest lucru a fost demonstrat la Orgatec de către Intel şi IBM într-o manieră impresionantă.

(Alfons Schräder/
Wolfgang Petersen)

Tendinţe

Aplicaţiile imaginabile ale tehnologiei DVI sînt diverse. Astfel, PC-ul multimedia va fi tot mai mult folosit în domenii precum comerţul, consultingul, informaţiile. Video-uri şi filme scurte, a căror desfăşurare poate fi comandată interactiv, sînt de valoare inestimablă în aceste domenii.

Şi în domeniul pregătire şi perfecţionare, multimedia va impulsiona utilizarea PC-urilor. Căci probabil că nu există cale mai eficientă de învăţare decît în această formă combinată.

Multimedia ascunde însă şi pericole, care în viitor nu vor putea fi ignorate. Deoarece informaţiile video, audio şi imaginile sînt prelucrate în formă digitală, posibilităţile de manipulare sînt mari.

Înregistrările video şi imaginile pierd mult din valoarea lor documentară, deoarece - după o manipulare - cu greu pot fi deosebite de o contrafacere.

Aşteptăm cu interes posibila funcţie a conceptului multimedia la depăşirea graniţelor între televiziune, calculator şi videotelefon. E posibil ca în viitor PC-ul să poată fi utilizat ca mediu universal în toate cele 3 domenii; ar putea fi posibile şi derulări simultane ale celor 3 aplicaţii.

Începutul a fost făcut cu tehnologia DVI. Şi deoarece Intel şi-a propus să integreze tot hard-ul sistemului într-un singur cip, în viitorul apropiat, care să poată fi implantat ca un coprocesor pe viitoarele plăci de bază, viitorul conceptului multimedia la o tot mai mare masă de utilizatori pare o certitudine.

Tehnică, Know-how

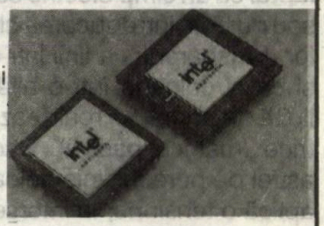
Una din problemele tehnologiei DVI este compresia volumului imens de date. Pentru aceasta, sistemul foloseşte un algoritm extraordinar de eficient. Acesta se bazează pe faptul că diferenţele între cadrele succesive ale unui film nu sînt mari. De aceea, se memorează numai diferenţele dintre cadre (delta frame compression).

De asemenea, este cunoscut faptul că ochiul omenesc nu este foarte sensibil la culori. Ca urmare, este posibilă memorarea numai a unei părţi din informaţiile de culoare, celelalte urmînd a fi refăcute ulterior prin interpolare. Prin acest procedeu, numit "chrominance subsampling", culorile şi luminozitatea nu sînt tratate ca şi componente RGB, ci ca şi componente Y-C. Y pentru luminozitate, luminanţă, iar C pentru tipul culorii (C1) şi saturaţia culorii (crominanţă - C2). Informaţiile de luminanţă sînt prelucrate la rezoluţia maximă, pe cînd la informaţia de culoare numai tot al 4-lea pixel este înregistrat. Cipurile DVI interpolează culoarea pixelilor lipsă la redare din culoarea pixelilor învecinaţi. Cît timp nu sînt prezente borduri tranşante între culori de saturaţie ridicată, acest principiu operează impecabil.

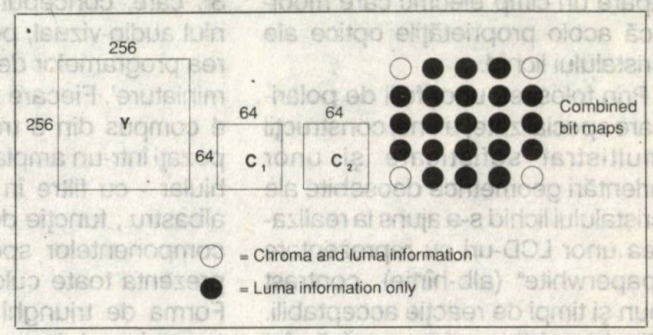
Compresia audio se face după algoritmul ADPCM (Adaptive Differential Pulse Code Modulation). Este vorba de un standard care se aplică şi la CD-ROM-XA.

Hard-ul DVI constă din două plăci - una de preluare, una de redare (capture board, delivery board). "Inima" hardului o constituie integratele A750PA şi A750DA. A750PA este procesorul de pixeli. Este vorba de un procesor microprogramabil, capabil să execute 12,5 milioane de operaţii pe secundă (MOPS). Nu conţine microcod permanent. De pe un Dual Port Video Ram (VRAM) microcodul este încărcat şi executat într-un RAM intern. A750Da este procesorul de afişaj. El lucrează ca şi super - CRT-controller în DVI şi nu poate fi adresat direct de procesorul gazdă; şi aici, programarea se face prin intermediul VRAM. Pe lîngă funcţia sa de controller CRT modulul realizează şi decomprimarea datelor video. Domeniul său special de aplicare este refacerea informaţiilor de culoare.

Inima hard-ului Intel A750



Procedura de compresie video "Chrominance Subsampling"



Pentru comanda hard-ului DVI, Intel a disponibilizat cîteva produse soft. Un driver TSR are grijă de comanda setului de cipuri DVI şi a procesorului audio. Pot fi utilizate cîteva biblioteci C la programarea subsistemului audiovizual (AVSS). Iar RTX asigură mutitasking-ul de timp real.

Ecranul viitorului: TFT - LCD

Königswinter, 23 august: Isamu Washizuka, Hiroshi Take şi Kozo Yano primesc premiul Eduard Rhein 1990. Toţi 3 sînt colaboratori ai laboratorului de cercetare şi proiectare al firmei Sharp Electronics din Japonia şi au avut contribuţii decisive la proiectarea unui ecran cu cristale lichide care să poată fi folosit pentru televiziune. Visul lui Eduard Rhein, televizorul de format mare, plat şi color, care să fie agăţat de perete ca un tablou, va fi în curînd realitate mulţumită tehnicii TFT.

Destul de repede inginerilor le-au devenit evidente limitele matricilor pasive de LCD-uri. Spre amintire: funcţionarea LCD-urilor se bazează pe faptul că lumina le poate traversa în mod diferit, funcţie de faptul că un cîmp electric este activ sau nu. Prin introducerea electrozilor transparenţi ca linii într-o placă şi ca şi coloane într-o altă placă, lipită de prima, se realizează o matrice (matrice pasivă). Dacă unei astfel de perechi linie-coloană i se aplică o tensiune, la intersecţia lor apare un cîmp electric care modifică acolo proprietăţile optice ale cristalului lichid.

Prin folosirea unor folii de polarizare specializate, a unei construcţii multistrat sofisticate şi unor orientări geometrice deosebite ale cristalului lichid s-a ajuns la realizarea unor LCD-uri cu reprezentare "paperwhite" (alb-hîrtie), contrast bun şi timpi de reacţie acceptabili. Display-uri tip matrice pasivă sînt produse de exemplu de Sharp, ca ecran color de 14", care la o rezoluţie de 640 x 480 pixeli poate reprezenta 512 culori.

Limitele display-ului matrice pasivă ies în evidenţă cînd se cere atît o rezoluţie mai bună cît şi un număr sporit de nuanţe de gri. În acest caz, contrastul cu greu atinge valori de peste 10 şi pînă la 15. Matricea activă rezolvă problema: aici valorile normale ale contrastului sînt de 70 pînă la 100! În acest caz, la fiecare intersecţie a conductorilor există un tranzistor de strat subţire - TFT (Thin Film Transistor). Punctele de pe ecran sînt comandate acum cu tensiuni sensibil mai mici, deoarece tranzistorii produc exact în locul pixel-ului un cîmp electric variabil.

Cam prin 1985, inginerii de la Sharp au început să examineze posibilităţile de realizare a unor monitoare cu cristale lichide în tehnologie TFT. În acel moment, încă nu existau în lume instalaţii care ar fi putut realiza aşa ceva. Dar tehnicile pentru producerea unor tranzistori atît de minusculi prin evaporare, implantare, corodare provin din microelectronică şi erau binecunoscute. Astfel că, pentru început, s-a folosit o instalaţie care servea pentru producerea unor plachete LSI de 6" care a fost transformată. Într-un fel, acela a fost momentul naşterii unei tehnici noi, numită astăzi "giant microelectronics".

Deja în 1987 primul ecran cu matrice activă produs de Sharp era pe piaţă. Era vorba de un TFT-LCD de 3", care, conceput pentru domeniul audio-vizual, permitea urmărirea programelor de televiziune 'en miniature'. Fiecare punct de ecran e compus din 3 tranzistori care - pozaţi într-un amplasament triunghiular - cu filtre în roşu, verde şi albastru , funcţie de luminozitatea componentelor specifice, pot reprezenta toate culorile spectrului. Forma de triunghi a dat numele acestui mod de amplasare al tranzistoarelor: configuraţie delta. Patru ani mai tîrziu a urmat un AMLCD (Activ Matrix LCD) de 4" şi în sfîrşit în 1989 un ecran TFT de 6". Şi acestea erau construite în configuraţie delta şi serveau unor aplicaţii video. Ca ecrane pentru calculatoare, AMLCD-uri în configuraţie delta pot fi folosite numai restrîns. Cu cît e mai mare un ecran (păstrînd rezoluţia) cu atît mai clar iese în evidenţă - mai ales la linii verticale - zigzagul amplasărilor triunghiulare. De aceea, un display TFT de 10" sau 14", care lucrează în standardul VGA, trebuie să aibă un alt mod de amplasare al tranzistorilor, pentru a anihila acest neajuns. Aici Sharp foloseşte configuraţia 'vertical stripe', în care tranzistorii sînt ordonaţi în rînduri şi coloane. PC-8501, laptop prezentat pentru prima dată la Orgatec 1990, are un ecran strălucitor de 10", care este un astfel de AMLCD în configuraţie vertical stripe. Sharp a început în 1990 să producă în serie mare TFT-LCD-uri de 10".

Nu este de mirare că aşa de puţini producători (ar mai trebui numiţi Hitachi şi Toshiba) pot produce aşa ceva, căci efortul tehnic şi economic pentru această producţie este imens. Un display color VGA necesită 640 x 480 x 3 - deci 921.000 - de tranzistori pe suportul de sticlă. Abaterile tranzistorilor unul faţă de altul trebuie să fie foarte mici. Nici unul nu are voie să fie 'ratat', deoarece în acel loc pe ecran ar apare o permanentă pată neagră. Comparînd producerea de display-uri TFT mari cu producerea de plachete de siliciu, se constată existenţa atît a unor asemănări, cît şi a unor deosebiri importante. Dacă pe plachetă cîteva cipuri sînt defecte, ele sînt izolate iar celelalte pot fi prelucrate în continuare. În 'giant microelectronics' însă, placheta este totdeauna un display întreg, care la o singură greşeală devine rebut. Ratele de rebut în această producţie sînt foarte mari şi sînt printre cele mai bine păzite secrete ale proiectanţilor. Numai Toshiba a făcut recent cunoscut că această rată 'este mult peste 90%'. Prin folosirea unei tehnici redundante, Sharp a putut să-şi organizeze producţia mai economicos. Un exemplu al acestui procedeu: tranzistori suplimentari îi pot înlocui pe cei defecţi.

Mărimea efortului pentru producţia AMLCD corespunde uriaşei provocări tehnice: Sharp şi-a propus să investească, pe parcursul a

3 ani, peste 1 miliard de mărci în tehnologia TFT.

Ce calitate a imaginii poate fi atinsă cu un AMLCD a fost arătat de Sharp la Electronica '90 în München. A putut fi văzut primul exemplar al unui display multimedia în format 10", folosibil atît în domeniul audiovizual cît şi în comunicaţii sau calculatoare ce poate reprezenta peste 16,7 milioane de culori! E vorba de un TFT-LCD în configuraţie vertical stripe şi cu rezoluţie VGA (640 x 3 x 480). În 1991, acest ecran, în format 10" şi 14" va intra în producţie de serie.

(Heimo Ponnath)

Tendinţe

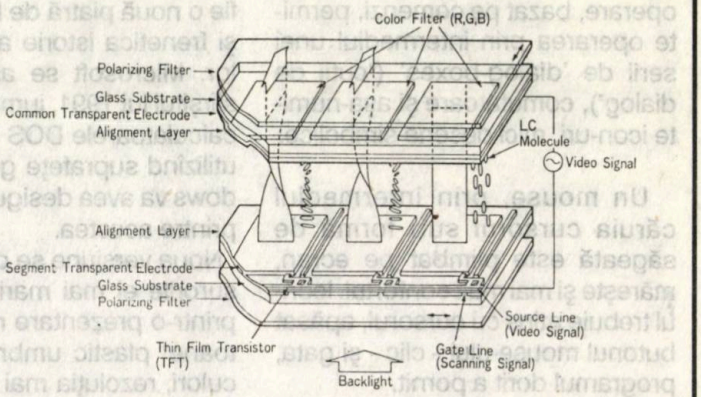
În momentul de faţă, tranzistorii tuturor TFT-LCD-urilor se depun cu siliciu amorf pe sticlă obişnuită, deosebit de netedă şi fără defecte. Acest material are avantajul că procesele litografice se desfăşoară la mai puţin de 300 grade Celsius, deci sub punctul de înmuiere al sticlei. Siliciul amorf are însă dezavantajul că mobilitatea electronilor în el este destul de redusă. O altă formă de siliciu, siliciul policristalin, are o mobilitate de 100 de ori mai mare a electronilor. Din acest fapt decurg două consecinţe interesante: pe de o parte, tranzistori din poli-Si comută semnificativ mai repede, pe de altă parte, s-ar putea mări curenţii prin ei, astfel că părţi ale logicii de comandă ar putea fi integrate pe suportul de sticlă. AMLCD-uri de acest tip ar necesita numai o fracţiune a conductorilor ce erau scoşi pînă acum din display-uri şi electronica anexă ar fi mult simplificată. Din păcate, utilizarea de poli-Si necesită încă pentru moment sticlă specială sau chiar cuarţ, deoarece temperaturile pentru producerea circuitelor sînt cam pe la 600-800 grade Celsius. În afară de aceasta, în acest caz conductorii şi elementele de comandă ocupă relativ mult loc, astfel încît relaţia între suprafaţa activă şi suprafaţa totală a unui display devine foarte nefavorabilă. Sharp lucrează la rezolvarea acestor probleme.

Pe moment, se realizează o altă cale: graţie unor conectoare flexibile, electronica de comandă este 'pliată' în spatele display-ului. Graţie realizării unor cipuri foarte plate, astfel de ecrane sînt încă destul de subţiri -cca. 2 cm - dar numărul conductorilor scoşi şi electronica de comandă suplimentară sînt mult reduse, astfel încît partea activă de ecran rămîne foarte mare.

Tehnică, Know-how

Probleme cu matricea pasivă apar la pretenţii crescînde relative la rezoluţie, mărimea pixelilor şi numărul de nivele de gri. Căci cîmpul electric nu apare numai la intersecţia liniei cu coloana, ci şi de-a lungul acestora. În afară de aceasta, cu cît este mai deasă reţeaua matricii, cu atît mai uşor apar efecte inductive în liniile sau coloanele alăturate (cross-talk). Dacă pe de altă parte se doreşte mărirea numărului de nivele de gri, cristalul lichid trebuie să reacţioneze tot mai uşor la schimbări mici ale cîmpului. Raportul de contrast la cele mai moderne display-uri TSTN este între 10 şi 15; ceea ce înseamnă că un punct de pe ecran la luminozitate maximă este de 10-15 ori mai luminos decît un punct complet stins. Soluţia problemei este matricea activă. La fiecare punct de intersecţie se găseşte un tranzistor minuscul, aplicat cu tehnologie de strat subţire pe sticlă. De aici numele de TFT - Thin Film Transistor. Un punct de ecran poate fi comandat acum cu o tensiune semnificativ mai mică; tranzistorul produce un cîmp electric variabil exact acolo. Pentru reglarea luminozităţii pixelilor se poate merge pe două căi în tehnica AMLCD. Prima constă în comandă directă prin mărimea cîmpului electric aplicat. O valoare mare a cîmpului duce atunci (la o orientare corespunzătoare a polarizatorului) la un pixel întunecat. Cu cît e mai mică intensitatea cîmpului, cu atît mai multă lumină trece prin cristalul lichid. Adunînd intensităţile luminoase ale celor 3 componente de culoare, se pot combina în acest fel foarte multe tonuri de culoare (de ex., 256.000 la display-ul de la PC-8501). Cerinţele pentru domeniul de toleranţă al tranzistoarelor sînt însă foarte mari în acest caz - aproape toate trebuie să aibă aproape aceeaşi caracteristică. Cealaltă cale e mai puţin pretenţioasă: aici tranzistorii trebuie să fie deschişi sau închişi. Prin impulsuri rapide se produce o succesiune de 'fulgere luminoase', pe care ochiul omenesc nu le deosebeşte decît ca valoare medie a luminozităţii. Cu cît este mai mare numărul de impulsuri pe unitatea de timp, cu atît mai luminos apare pixelul. Şi aici componentele culorii de intensităţi diferite se adună, determinînd culoarea aparentă a punctului. Dezavantajos în acest mod de lucru este pe de o parte faptul că este necesar un efort mai mare pentru comanda display-ului, iar pe de altă parte că prin regimurile tranzitorii datorate impulsurilor rapide, chiar şi la curenţi mici apar inducţii care micşorează contrastul. Probabil că în curînd şi aici vor fi atinse limitele ce pot fi bănuite şi care şi la display-urile cu matrice pasivă îngrădesc rezoluţia. AMLCD-urile de acest tip pot ajunge de aceea la valori ale contrastului de pînă la 60:1, pe cînd altfel valori de 100:1 sînt uzuale.

Schema de funcţionare a unui TFT-LCD



Ferestre în lumea datelor

Cu noua versiune de Windows, Microsoft a demonstrat odată în plus cine are cel mai important cuvînt de spus cînd este vorba despre tehnica ferestrelor în domeniul PC-urilor.

Anii 80 marchează cea mai importantă revoluţie din istoria calculatoarelor. Acest deceniu a reuşit să producă pentru prima dată, graţie progreselor miniaturizării şi avantajelor de preţ posibile datorită producţiei în serie mare, calculatoare personale şi de casă pe care, pentru prima dată, putea întradevăr să şi le permită oricine. La fel de importantă însă ca avantajele de spaţiu şi preţ trebuie considerată o a doua realizare a acestui deceniu, ascunsă sub anodinul acronim GUI. Prescurtarea vine de la "Graphical User Interface" şi desemnează o interfaţă cu utilizatorul bazată pe ferestre, care spre deosebire de promptul sistemului de operare, bazat pe comenzi, permite operarea prin intermediul unei serii de 'dialog-boxes' ("cutii de dialog"), comutatoare şi aşa-numite icon-uri, mici desene simbolice.

Un mouse, prin intermediul căruia cursorul sub formă de săgeată este plimbat pe ecran, măreşte şi mai mult confortul. Iconul trebuie atins cu cursorul, apăsat butonul mouse-ului - clic - şi gata, programul dorit a pornit.

Primul calculator care a pus în practică acest principiu, "klickn' roll", a fost Lisa (Apple). Deşi calculatorul ca atare s-a dovedit a fi un eşec, suprafaţa lui utilizator prietenoasă a fost primită cu entuziasm atît de specialişti cît şi de utilizatori. La începutul anului 1984 a urmat Apple Macintosh, căruia, fără îndoială, îi revine meritul de a fi făcut populare suprafeţele grafice. Utilizatorii DOS în schimb au trebuit să se uite cu invidie la proprietarii de Macintosh pînă la sfîrşitul anului 1984, cînd Microsoft a deschis compatibilelor fereastra spre date cu Windows 1.0.

Aplicaţiile orientate spre grafică, (desenare, CAD, DTP) au constituit desigur partea leului a primelor aplicaţii. În ciuda aprecierii unanime pe care noul principiu a găsit-o datorită uşurinţei sale în folosire, acceptanţa lui a rămas surprinzător de redusă în prima fază. Ca orice inovaţie, şi tehnica ferestrelor trebuia întîi să-şi cucerească locul.

Pînă azi încă, marea majoritate a utilizatorilor sînt de părere că ajung mai repede la obiectivul dorit prin intermediul aparent mai complicat al prompter-ului. Căci suprafaţa utilizator, cu 'balastul' ei grafic, influenţează negativ timpul de răspuns al sistemului.

Totuşi, conform dorinţei creatorilor săi, Windows 3.0 urmează să fie o nouă piatră de hotar în scurta şi frenetica istorie a calculatoarelor. Microsoft se aşteaptă ca la sfîrşitul lui 1991, jumătate din toate calculatoarele DOS să fie operate utilizînd suprafeţe grafice. Şi Windows va avea desigur partea leului printre acestea.

Noua versiune se deosebeşte de surorile ei mai mari întîi de toate printr-o prezentare mai 'la zi'. "Butoane" plastic umbrite, mai multe culori, rezoluţia mai înaltă precum şi icon-uri mai multe şi mai viu colorate corespund spiritului vremii şi subliniază înrudirea cu Presentation Manager de sub OS/2. Ca o supra-găselniţă, fiecare poate să-şi creeze un fundal individualizat, care, ca grafică bitmap, se constituie în fondul pe care apoi defilează tot peisajul de ferestre.

O îmbunătăţire reală este forma nouă a rutinei de instalare. Modificări pot fi făcute acum fără probleme prin intermediul icon-ului 'Windows-Setup', care de altfel preia şi instalarea automată a aplicaţiilor pe care Windows le recunoaşte. Au trecut vremurile cînd orice modificare însemna o reinstalare completă, care să necesite o trecere prin toate dischetele de instalare.

Schimbul fără probleme de text şi grafică prin intermediul clipboard-ului, ca şi funcţia lui de 'cut-and-paste' (taie şi păstrează) este asigurat datorită definiţiilor exacte de format. Transferul de date între diferite aplicaţii devine astfel o joacă de copil, avantaj pe care utilizatorii de Apple Macintosh îl ştiu preţui deja de cîţiva ani.

Cele mai importante modificări au avut loc însă 'dedesubt': gestiunea memoriei este adaptată noilor realizări din domeniul PC-urilor. Pe 386 Windows face posibil şi multitaskingul, iar pe 286 ştie să folosească fără probleme memoria suplimentară. Prin capacitatea de a folosi modul de lucru protejat (Protected Mode) pe AT-uri, termină definitiv cu una din cele mai deranjante limitări din DOS: limita celor 640 kbytes. În modul 386, unei aplicaţii îi sînt oferite chiar pînă la 16 Mbyte.

Practic nici un avantaj nu obţin însă posesorii de XT-uri - ei vor simţi numai pierderea de viteză a 'reacţiilor' sistemului. În rest, totul rămîne cum a fost, căci modul de lucru real (Real-Mode) în care trebuie să ruleze Windows pe un 8088 sau 8086 nu lasă loc pentru nici un fel de îmbunătăţiri în gestiunea memoriei.

Ramura de soft s-a acordat mai repede ca oricînd cu inovaţia. Aplicaţii care în prima fază au 'mizat pe alţi cai', (ca de exemplu Ventura Publisher), au fost portate sub Windows cu iuţeala gîndului. La ele s-au adăugat actualizări la aplicaţii mai vechi ca şi aplicaţii cu totul noi, astfel încît deja la mai puţin de jumătate de an după prezentarea noii versiuni sînt disponibile programe şi utilitare din toate domeniile. După 'atacul' purtat împotriva

GEM-ului şi pe care Windows pare să-l fi cîştigat, deja se întrevăd confruntări cu noi concurenţi; printre ei Presentation Manager de sub OS/2 din 'grajdul propriu', dar şi outsideri de tipul unui "GEOS", născut pe C64. Pe moment însă, poziţia de lider a lui Windows 3.0 este de necontestat.

(Charles Climm)

Tendinţe

Tendinţa spre suprafeţe utilizator grafice este vizibilă în întregul peisaj calculatoristic şi probabil că nu mai poate fi oprită. Chiar şi lumea Unix-ului se lasă cucerită de mijlocitorii unificanţi între om şi maşină, comandaţi cu ajutorul unui mouse. Anii 90 probabil că nu vor mai cunoaşte proiecte de calculatoare care să nu fie din start orientate pe tehnica de ferestre şi mouse.

Suprafaţa utilizator prietenoasă va duce la micşorarea reţinerilor din partea utilizatorilor. Căci pe de o parte solicitările de opţiuni deseori complicate dinspre sistemul de operare îi sînt 'ascunse' utilizatorului, iar pe de altă parte se scurtează timpii de învăţare, deoarece elementele de comandă ale tuturor programelor se uniformizează şi trebuie învăţate o singură dată. Cine plimbă cu mouse-ul o săgeată pe ecranul multicolor şi poate opera astfel toate funcţiile unui program, scapă repede de orice reţinere în lucrul cu calculatorul.

Dacă ar fi să aducem şi un ton mai grav în discuţie, s-ar putea vorbi de 'dublul tăiş' al acestei 'săbii'. Căci avantajul faptului că tot mai mulţi oameni pot umbla tot mai uşor cu calculatoarele conţine totodată şi dezavantajul că tot mai puţini utilizatori vor avea o imagine reală despre ce se întîmplă înăuntrul calculatorului lor.

Pe de altă parte, sigur că nu e necesar să şii detalii despre funcţionarea motorului Otto ca să fii un bun automobilist. Astfel, unei destinaţii calculatorul i se apropie în acest fel tot mai mult: cea de a fi o unealtă universală pentru rezolvarea rapidă a aproape oricărei probleme imaginabile.

Vom constata că în viitor şi programarea, cu cerinţele ei complexe, va fi din ce în ce mai mult o chestiune de "klick'n roll" - plimbat cursorul şi apăsat butonul mouse-ului. Primele tentative sub Windows 3.0, sisteme de proiectare hypercard, ca Object Script, urmează exemplul statuat de Next-Step. Părintele "Lisa"-ei, Steven Jobs, a utilat suprafaţa utilizator a Next-ului de la bun început cu un 'Interface Builder' - un "constructor de interfeţe". Acesta permite asamblarea de programe noi din module existente la o simplă bătaie din palme - un 'clic' pe butonul mouse-ului.

Cele mai importante tendinţe pentru anii '90 se prefigurează sub noţiunile 'multimedia' şi 'workgrouping'. Pentru dezvoltările acestora, suprafeţele grafice vor fi platforma determinantă.

Tehnică, Know-how

Cel mai simplu o fereastră se defineşte ca fiind o zonă a ecranului, delimitată de un cadru, a cărei dimensiuni pot fi modificate şi care poate fi deplasată pe ecran. Barele de rulaj ("rollbars") de la marginile ferestrei permit "frunzărirea" atît în sus şi în jos, cît şi stînga - dreapta. De obicei sînt completate de butoane de apăsare ("push-buttons"), care la un clic al mouse-ului închid fereastra, o micşorează transformînd-o într-un icon, un simbol a conţinutului sau rolului ei, sau o măresc la dimensiunea întregului ecran.

Dialogul cu programul se face prin intermediul unor elemente grafice. De regulă acestea sînt icon-uri, imagini-simbol pentru anumite funcţii care sînt activate odată cu un clic asupra lor. Cutiile de dialog sînt ferestre mai mici, care se deschid în timpul derulării unui program şi de regulă conţin o întrebare la care utilizatorul trebuie să răspundă prin "da" sau "nu"; răspunsurile posibile sînt prezente şi ele în cutia de dialog; cel dorit trebuie selectat, adică: cursorul trebuie poziţionat cu ajutorul mouse-ului deasupra lui, după care trebuie apăsat butonul mouse-ului (clic...).

Meniurile - fie ele "pop-up" sau "push-down" - sînt meniuri ce stau frumos ascunse, locul lor fiind marcat într-un rînd de meniuri de un simbol (icon) specific; abia clicul (clicuirea? clicarea? clicajul?) asupra acestui simbol determină expandarea lui la mărimea normală, permiţînd inspectarea/modificarea tuturor opţiunilor posibile din meniu.

Acest mod de abordare se deosebeşte radical de mediile orientate spre linii de comandă.

Apariţia lui este legată de primele cercetări serioase asupra interfeţelor om-maşină, efectuate la începutul anilor 70 la Palo Alto Research Center.

Apple şi Microsoft au fost primele firme care s-au ocupat de transpunerea conceptului în practică la nivelul PC-urilor. Cu apariţia lui Windows 3.0 s-a atins un stadiu de dezvoltare care marchează stadiul actual al tehnicii şi care face ca DOS - PC - urile să fie în urma Macintosh-urilor numai în ceea ce priveşte viteza de execuţie.

În locul versiunilor distincte pentru 8086, 80286 şi 80386 există acum din nou o versiune unică, care funcţie de contextul hard poate rula în trei moduri distincte. În modul real se realizează compatibilitatea maximă cu aplicaţii mai vechi, în modul standard şi pe 80286 se poate accesa memoria suplimentară. Modul 386 oferă cele mai largi posibilităţi, cu exploatarea integrală a capacităţii de memorie existente, exploatarea memoriei virtuale şi posibilităţilor de multitasking. Programele DOS oarecare pot fi executate în multitasking într-o "fereastră DOS".

Elementele grafice au fost "stilate" şi oferă toate facilităţile aşteptate de la o suprafaţă grafică. Reprezentarea a fost făcută pe cît posibil independent de hard-ul de grafică, asfel încît în limite destul de largi aspectul este aproape acelaşi.

Schimbul de date între diferite aplicaţii este simplu, graţie formatelor strict formalizate.

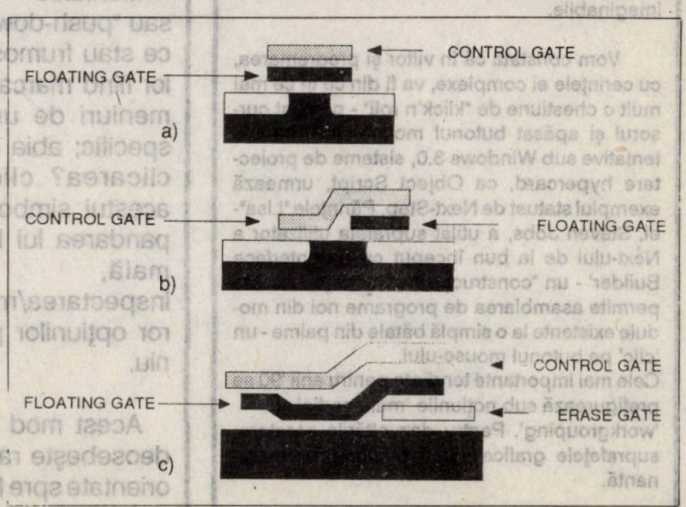
Următoarele extensii vizate de Microsoft cuprind folosirea standardizată de surse audio şi video pentru aplicaţii multimedia.

Tehnologia flash
Mediu de memorare de viitor

Cu producerea în serie a EPROM-ului 'nevolatil' ETOX-II de către Intel pare să se fi împlinit reuşita unui nou mediu de memorare.

Întreaga piaţă a calculatoarelor portabile - de la laptop pînă la palmtop - poate profita de această nouă tehnologie. Plăcile de memorie flash ce pot fi rescrise sînt mai mici decît dischetele de 3,5", nu se pot uza fizic şi nu necesită o unitate, ci sînt adresate direct, printr-o interfaţă specială ce foloseşte 68 de pini. Ceea ce înseamnă că durata de funcţionare a unui laptop sau notebook poate fi mărită considerabil, acesta poate deveni mult mai uşor datorită eliminării unităţilor de disc iar da e sensibile sînt păstrate mult mai sigur pe un suport ermetic închis şi nedegradabil cum este placa de memorie flash. Capacitatea ridicată, pînă la 4 Mbyte, pledează şi ea pentru tehnologia flash. Ea nu va influenţa însă numai piaţa calculatoarelor portabile. Şi pentru PC-uri folosite în industrie, sau şi în staţii de lucru (workstations), tehnologia este interesantă prin cîteva din caracteristicile ei. Plăci de extensie pot fi dotate cu capacităţi de memorare de pînă la 16 Mbytes, în curînd chiar 32 Mbytes. Deoarece la modulele implementate pe placa de AT e vorba de acelaşi principiu tehnic ca la plăcile de memorie, este asigurată o înaltă securitate a datelor chiar şi într-o incintă supusă la efecte fizice puternice.

Ideea tehnologiei flash nu este nouă. Deja prin inventarea şi producţia EPROM-ului, în 1972, de către Intel, fundamentul pentru mediul semiconductor nevolatil era creat. Alţi cercetători au perfecţionat metoda şi, în 1986, Seeq Technology a scos pe piaţă nMOS-ul, o memorie semiconductoare concepută pe principiul split-gate. Acest cip era un flash-EPROM de 512 kbit, dar din cauza modului său de construcţie era cu circa 10% mai mare decît un EPROM uzual. Concernul multinaţional Toshiba a încercat să valorifice un concept propriu şi a construit o memorie flash nMOS de 256 kbiţi, care pe deasupra corpului de oxid necesita 3 nivele de structură, lucru care a avut ca efect faptul că şi acest cip a ieşit prea mare. Şi Hitachi lucra în domeniul tehnologiei flash şi s-a concentrat în principal pe dezvoltarea unui cip de 1Mbit. Exact ca la Intel, s-a mers pe ideea unui singur tranzistor, integrat într-o celulă. Deosebirea faţă de conceptul Intel constă în faptul că la Hitachi trebuie să străbată două domenii difuze pînă la şi de la floating gate. În acest fel, Hitachi încerca să excludă o descărcare a mediului de memorare, aşa-numitul 'curent de pierdere'. Reuşita a aparţinut pînă la urmă Intel-ului, cu elaborarea lui ETOX-II, care permite realmente o producţie în serie şi deci o valorificare comercială. Între timp, tehnologia Intel este acceptată ca standard de cei mai importanţi producători de hard şi soft. Şi SCM, singurul producător în Germania în domeniul plăcilor de memorie flash şi al discurilor flash s-a decis pentru conceptul Intel. Tehnologia flash lucrează în principiu ca un EPROM, cu cîteva deosebiri mici, dar marcante. EPROM-urile pot fi şterse şi rescrise extern, folosind lămpi cu raze ultraviolete şi respectiv echipamente de programare. În tehnologia flash, atît ştergerea cît şi scrierea trebuie să se desfăşoare ca ciclu intern şi trebuie realizată o densitate mult mai mare a tranzistorilor individuali. Deabia acoperiri oxidice inovative şi progrese în fotolitografie au dus la realizarea tranzistorului individual dublu-polisiliconic (double- polysilicon single-transistor) de către Intel. Astfel se pot desfăşura peste 10.000 de cicluri de citire şi scriere într-un singur cip şi se poate atinge o densitate optimă a tranzistorilor individuali. ETOX-II al Intel-ului este o memorie flash de 512 kbiţi şi, datorită modului ei de realizare, ermetic închis, memoria semiconductoare care se recomandă singură pentru viitor. Pînă acum, conceptul Intel al tehnologiei flash pare să se consacre, deoarece alte concepte şi realizări, (ca de exemplu cea de la Toshiba), nu au putut fi valorificate comercial pînă la această dată. Dacă vînzările de calculatoare portabile se menţin ridicate, utilizarea tehnologiei flash nu va determina numai o cădere a



Diferitele concepte în tehnologia flash:
a) Intel
b) Seek
c) Toshiba

preţurilor la laptop-urile uzuale, dar va contribui şi la îmbunătăţirea ergonomiei calculatoarelor portabile, va mări capacitatea de memorare a suporţilor de date de la 1Mbyte la 4 Mbyte şi va permite creşterea substanţială a timpului de funcţionare al acestor aparate.

(Reiner König)

Tendinţe

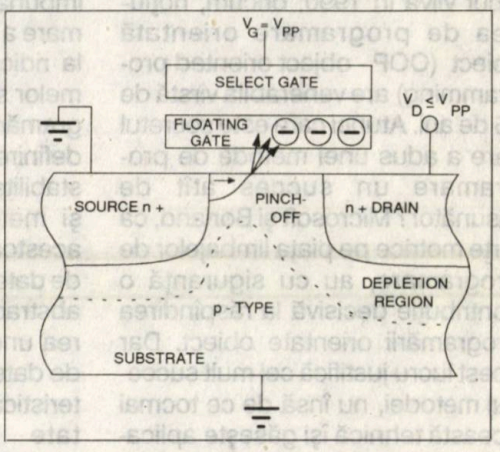
Miercuri, 22 august 1990, Double Free Hotel undeva în Santa Clara, California. Tot ce are rang sau nume în lumea calculatoarelor se întîlneşte în această zi în acest hotel. Lista de oaspeţi permite evaluarea semnificaţiei acestei întîlniri. Intel, IBM, Compaq, Hewlett-Packard, Lotus, Microsoft, Wordperfect, Hitachi, Toshiba, Fujitsu şi mulţi alţii. Au venit cu toţii pentru a 'lansa la apă' noul standard. A avut loc aşa-numita PC Memory Card International Association, pe scurt PCMCIA. S-a convenit asupra faptului că pentru o valorificare comercială a cartelelor de memorie (memory cards) trebuie să existe un standard obligatoriu. Această premisă a fost creată la 22 august. Toţi producătorii de hard şi soft prezenţi au acceptat caracteristicile prezentate şi discutate. Astfel încît nu mai exista nici o piedică în calea producţiei în serie. Înainte de reglementarea unui standard tehnic toţi producătorii au trebuit să creeze condiţiile cadru pentru producţia de cartele de memorie. Superputerile "electronice" Japonia şi SUA au ajuns la un acord în următoarele puncte: - o capacitate de adresare de pînă la 64 Mbyte - bus de date de 16 sau 8 biţi - reglementarea suportului pentru memorii flash - PCMCIA şi grupul japonez de standardizare JEIDA îşi acordează reciproc specificaţiile pentru cartele de memorie.

De asemenea au fost stabilite toate caracteristicile determinante pentru imaginea cartelelor de memorie. Vor exista cartele de memorie de tip I şi de tip II; ele vor avea aceeaşi lungime şi lăţime, dar se vor deosebi prin grosime. Racordarea la placa PC-ului se va face printr-un conector normat de 68 de pini. De-abia o jumătate de an a trecut de la această întrunire; crearea unor noi locuri de producţie pentru producţia de cartele de memorie va mai dura un timp. Însă un lucru devine evident de pe acum: producţia de serie a cartelelor de memorie va fi începutul sfîrşitului pentru unităţile de discuri, consumatoare de spaţiu şi energie.

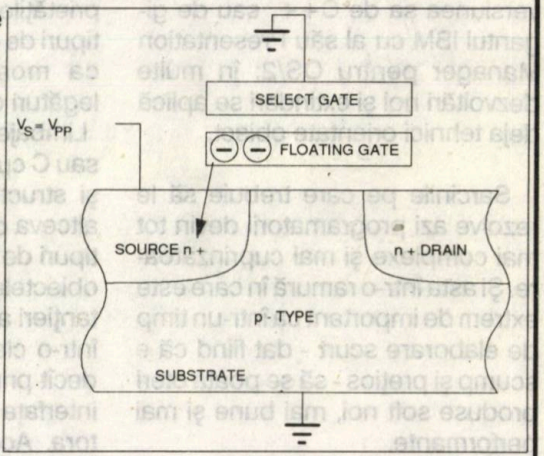
Tehnica, know-how

Cipul de memorie flash ETOX-II conceput de Intel foloseşte o folie dublustrat polysilicon -n -well - CMOS de 1,0m. Memoria flash internă de 65.536 x 8 biţi foloseşte un tranzistor individual de 3,8 mm x 4,0 mm. În modul programare, o memorie flash se comportă exact ca o memorie EPROM convenţională. O tensiune ridicată generează electroni care curg printr-un spaţiu intermediar.

Tensiunea ridicată de pe "control gate" (poarta de control) scoate electronii liberi din oxid gate şi îi fixează pe floating gate. Este exact acelaşi lucru care se întîmplă într-un EPROM, cu singura diferenţă că distanţa dintre floating gate şi corpul oxidului, de circa 100 Ang strömi, este numai aproximativ o treime din valoarea acestei distanţe la un EPROM uzual. Diferenţa determinantă apare la ştergerea celulei de memorie flash. EPROM-ul necesită lumină ultravioletă pentru a neutraliza înregistrările în floating gate. La o memorie flash cum este ETOX-II, un cîmp electric trece pe deasupra corpului de oxid, eliberînd electronii legaţi în floating gate şi realizînd astfel ştergerea. Acest mecanism de ştergere este o adaptare EPROM care funcţionează după principiul "Fowler-Nordheim". Memoria de 512 kbiţi a Intelului lucrează cu o aşa-numită arhitectură cu port de comandă (command port architecture), care simplifică controlul microprocesorului pe toată durata citirii, ştergerii sau înscrierii.



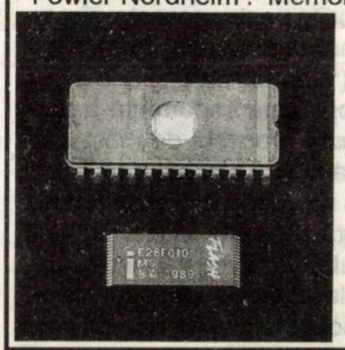
Celula de memorie flash ETOX în timpul programării



Aceeaşi celulă de memorie flash în timpul ştergerii

Acest port de comandă constă dintr-un registru de comenzi, un decodor de comenzi, un latch de stare precum şi două latch-uri pentru date şi adrese.

Dacă la intrarea de 12V nu este prezentă tensiunea de 12V, se execută operaţii normale de citire. Dacă însă se aplică tensiunea de 12V, portul de comandă iniţiază procesele de ştergere sau programare.



Noile tehnologii de fabricaţie permit saltul de la "EPROM-ul epocii de piatră" la cipurile flash

Programare orientată obiect
Transparenţă prin "clase"

Nu e o tehnică nouă cea care a făcut vîlvă în 1990; oricum, noţiunea de programare orientată obiect (OOP - object oriented programming) are venerabila vîrstă de 25 de ani. Atunci care este secretul care a adus unei metode de programare un succes atît de răsunător? Microsoft şi Borland, ca forţe motrice pe piaţa limbajelor de programare, au cu siguranţă o contribuţie decisivă la răspîndirea programării orientate obiect. Dar acest lucru justifică cel mult succesul metodei, nu însă de ce tocmai această tehnică îşi găseşte aplicare într-un limbaj de nivel înalt. Că e vorba de firme mici ca Zortech, cu versiunea sa de C++, sau de gigantul IBM cu al său Presentation Manager pentru OS/2: în multe dezvoltări noi şi extinderi se aplică deja tehnici orientate obiect.

Sarcinile pe care trebuie să le rezolve azi programatorii devin tot mai complexe şi mai cuprinzătoare. Şi asta într-o ramură în care este extrem de important ca într-un timp de elaborare scurt - dat fiind că e scump şi preţios - să se poată oferi produse soft noi, mai bune şi mai performante.

Posibilitatea ca la elaborarea unui produs nou să se poată folosi ceva deja existent, verificat şi probat în practică, de a modifica şi dezvolta acest ceva, ca şi tendinţele de standardizare la suprafeţele utilizator sau în softul de comunicaţii sînt adevăratele forţe motrice care au dus la răspîndirea programării orientate obiect.

Programarea orientată obiect nu este numai o tehnică de programare, pentru a refolosi lucruri deja verificate şi încercate, ci este şi o metodă prin care programarea structurată poate fi rafinată şi îmbunătăţită. In afară de aceasta, OOP se pretează foarte bine la îmbunătăţirea capacităţii de exprimare a limbajelor de programare şi la ridicarea transparenţei programelor scrise. Ideea de bază a programării orientate obiect constă în definirea de obiecte, în care sînt stabilite atît cele mai diferite date cît şi metodele pentru prelucrarea acestora. Amestecul astfel născut, de date şi funcţii, se denumeşte tip abstract de date sau clasă. Formarea unor astfel de tipuri abstracte de date sau clase nu este o caracteristică proprie programării orientate obiect. Sub numele de "modul", în Modula-2 se înţelege acelaşi lucru. Tipice sînt proprietăţile pe care le prezintă aceste tipuri de date, denumite cu termeni ca moştenire, polimorfie sau legături dinamice.

Limbaje de programare ca Pascal sau C cunosc tipuri de date simple şi structurate. Variabilele nu sînt altceva decît elemente ale acestor tipuri de date. La fel se întîmplă cu obiectele. Nu sînt altceva decît instanţieri ale claselor. Datele definite într-o clasă nu mai pot fi utilizate decît printr-un aşa-numit protocol, interfaţa "oficială" cu lumea a acestora. Acest proces, de a permite manipularea datelor numai prin intermediul unor operaţii permise pentru aceasta, poartă în programarea orientată obiect numele de încapsulare (encapsulation). Cum anume sînt implementate rutinele individuale, ce operaţii execută de fapt şi ce structuri interne de date sînt folosite pentru aceasta, sînt lucruri care utilizatorului îi rămîn ascunse ("data hiding").

O proprietate importantă a unei clase este procesul de moştenire (inheritance). Se poate deci defini o clasă nouă pornind de la o clasă existentă. Această clasă derivată sau subclasă "moşteneşte" proprietăţile clasei superioare. Subclasa poate lărgi structura de date moştenită, poate modifica rutinele moştenite sau chiar să primească noi rutine. Cu acest mecanism de transmisie ereditară este posibilă refolosirea codului de program deja folosit şi testat. Căci în subclasă nu mai trebuie programat decît ceea ce s-a adăugat nou sau s-a modificat faţă de clasa superioară. Există astfel posibilitatea de a depune în biblioteci ierarhii întregi de clase, în care clasele sînt caracterizate prin relaţii ereditare.

Activarea unui obiect se poate face prin intermediul unui "mesaj", restul rămîne apoi în sarcina exclusivă a obiectului. Iar obiecte din clase diferite pot fi adresate cu un acelaşi mesaj. Semnificaţia mesajului se stabileşte însă abia în legătură cu numele clasei. De exemplu, prin intermediul limbajului, stări de fapt diferite pot fi exprimate cu ajutorul unui acelaşi mesaj. Spre exemplu, în frazele "gazda îşi conduce musafirii" şi "sîrma de cupru conduce curentul electric", cuvîntul "conduce" exprimă de fiecare dată altceva. Semnificaţia mesajului "conduce" devine clară abia în legătură cu obiectele. Această caracteristică a unor clase diferite de a reacţiona potrivit specificului lor la mesaje identice se numeşte polimorfism. Polimorfismul poate fi utilizat pentru a crea programe mai transparente, mai uşor lizibile.

Pentru utilizator, a programa orientat obiect înseamnă a prelua ceva ce a fost deja testat, a modifica acest lucru potrivit noilor cerinţe şi a-l extinde. Acest lucru este valabil îndeosebi în domenii în care există sau sînt pe cale de a se afirma standarde soft. Astfel de exemplu standardul SAA stabileşte cîteva caracteristici pentru suprafeţe utilizator. Mediul de dezvoltare din Turbo-Pascal 6.0 - numit şi Turbo-Vision - este programat complet orientat obiect.

Mai mult: toate obiectele folosite pentru Turbo-Vision îi sînt puse la dispoziţie utilizatorului pentru aplicaţiile proprii. Astfel, apelînd numai la soluţii verificate în practică, programatorul poate totodată să se ţină de un standard care s-a impus.

Peste tot unde se caută aplicaţii standardizate, ca de exemplu la construcţia de sisteme de comunicaţie care să poată schimba date în cele mai diferite forme (texte, grafică, imagini) bibliotecile de obiecte îşi găsesc aplicarea.

Dar nu numai aplicaţiile standardizate sînt locul în care îşi găsesc de lucru programatorii de obiecte, căci şi dezvoltarea unor aplicaţii deja existente se pretează foarte bine pentru OOP.

Cu OOP, programarea structurată se îmbogăţeşte cu încă o metodă. Caracteristicile ei determină o îmbunătăţire calitativă a softului, avînd în final efecte benefice atît în ce priveşte timpul necesar elaborării cît şi în ce priveşte costul.

OOP nu numai că va găsi o largă răspîndire în aplicaţii standardizate, ci va contribui implicit la accentuarea tendinţelor de standardizare.

(Andreas Schimpf)

Tehnică, Know-how

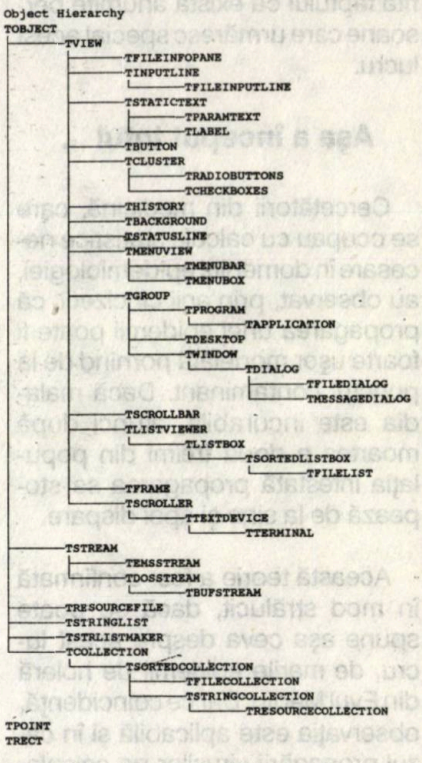
Programarea orientată obiect nu este altceva decît o nouă metodă de a scrie programe structurate, transparente şi eficiente. Răspîndirea ei tot mai mare se datorează cerinţelor crescute către diversele aplicaţii. În anii ce vin, OOP va fi o tehnică de programare care va trebui să aparţină instrumentarului oricărui bun programator, indiferent că-şi scrie aplicaţiile în C, Pascal sau Smalltalk.

Pe măsură ce se vor modifica cerinţele faţă de soft, se vor dezvolta şi metodele şi prin aceasta şi OOP. În primul rînd necesitatea unor timpi de elaborare scurţi, împreună cu cea a unor programe puternice şi confortabile, dar totodată ieftine, au determinat decizia caselor de soft de a introduce pe scară largă OOP.

Multe din limbajele mai mult sau mai puţin răspîndite azi vor prezenta numai interes istoric peste cîţiva ani. Dar revelaţiile pe care acestea le-au permis vor intra în cele cîteva limbaje orientate obiect care vor domina piaţa. Printre ele, desigur şi Pascal şi C++, ultimul în corelaţie cu viitorul sistemului de operare Unix.

În prima fază însă, metoda va contribui în primul rînd la impunerea anumitor standarde în această ramură. Mai ales pentru realizarea celor trei sisteme de reguli ale SAA programarea orientată obiect este indicată prin excelenţă. Pentru definiţiile de interfaţă, liniile directoare pentru suprafeţe utilizator şi sistemele de comunicaţii fixate în SAA vor exista în curînd biblioteci cuprinzătoare şi performante.

Deja există încercări de rezolvare pentru cele mai diverse limbaje de programare. Astfel, s-au impus unanim cele stabilite relativ la o ghidare unitară a utilizatorului. Prin OOP, modificarea şi lărgirea acestor biblioteci se face mai simplu decît prin metode convenţionale. Căci codul bun, verificat şi selecţionat, se 'moşteneşte', astfel că aplicaţii noi sînt mai sigure şi mai puţin afectate de erori.



Ierarhia obiectelor la Turbo Vision. Tot ceea ce poate fi reprezentat, trebuie să fie un urmaş al lui TView.

Tendinţe

Simula este "mama" tuturor limbajelor orientate obiect. Ea fost derivată pe la mijlocul anilor '60, în Computer-Zentrum din Oslo, din mai vechiul Algol. Simula era gîndit pentru simulări.

Ideea că, pentru o operare uşoară a calculatorului, o suprafaţă utilizator grafică puternică ar putea fi de importanţă determinantă, a dus la apariţia limbajului Smalltalk, la sfîrşitul anilor '60. Totuşi, Smalltalk a fost făcut cunoscut publicului larg abia în 1981, prin intermediul revistei BYTE. Apoi, tehnici de programare orientate obiect au intrat tot mai mult în diverse limbaje de programare. Şi pentru Pascal există extensii orientate obiect de mai mult timp. Ca de exemplu Clascal, înlocuit mai tîrziu de Object-Pascal pe calculatoarele Apple.

De la mijlocul anului 1989, cu versiunile 5.5 de Turbo-Pascal şi 1.0 de Quick-Pascal de la Borland şi respectiv Microsoft există versiuni de Pascal orientate obiect pe PC-uri.

Dar deja mai de mult, la începutul anilor '80, a apărut C++; C este un subset adevărat al C++-ului. Astăzi, C++ există ca preprocesor de la cei mai diverşi producători, pentru sistemele de operare Unix şi MS-DOS.

Dezvoltarea programării orientate obiect a avut loc, în trecut, în afara 'curentului principal' al programării. Fundamentele au fost puse la sfîrşitul anilor '60 şi începutul anilor '70. De-abia în a doua jumătate a anilor '80 programarea orientată obiect a intrat în conştiinţa masei mari a programatorilor.

Fascinaţia viruşilor

Partea cea mai valoroasă a unui sistem de calcul o constituie datele. Indiferent că este vorba de tabele, texte editate, surse ale unor programe proprii, baze de date sau chiar de contabilitatea unei firme, oricare ar fi conţinutul lor, comparativ cu valoarea datelor valoarea oricărui hard sau soft devine nesemnificativă. Din păcate ele sînt expuse pericolului de a fi deteriorate nu numai datorită unor defecţiuni tehnice, sau datorită unor erori în programe, ci şi datorită faptului că există anumite persoane care urmăresc special acest lucru.

Aşa a început totul ...

Cercetătorii din medicină, care se ocupau cu calcule statistice necesare în domeniul epidemiologiei, au observat, prin anii cincizeci, că propagarea unei epidemii poate fi foarte uşor modelată pornind de la punctul contaminant. Dacă maladia este incurabilă, atunci după moartea a două treimi din populaţia infestată propagarea se stopează de la sine şi apoi dispare.

Această teorie a fost confirmată în mod strălucit, dacă se poate spune aşa ceva despre acest lucru, de marile epidemii de holeră din Evul Mediu. Dar ce coincidenţă, observaţia este aplicabilă şi în cazul propagării viruşilor pe calculatoare.

Viruşii, ca programe, au apărut, ca noţiune, în publicaţiile de specialitate, în 1974, în studiul ACM "Use of Virus Functions to Provide a Virtual APL Interpreter under User Control" de B. Gunn. Alte studii referitoare la acelaşi subiect au apărut la începutul anilor '80. În acea perioadă cercetările se făceau în cel mai strict secret, deoarece unele cercuri militare considerau că programele virus sînt utilizabile pentru copierea şi furtul unor tehnologii şi softuri şi pentru paralizarea sistemelor de calcul ale puterilor inamice (idee care n-a dispărut încă !). Existenţa acestor programe a devenit publică abia în 1984 cînd revista vest-germană Spiegel a relatat, într-un scurt articol, despre existenţa programelor autoreproductibile şi a atras atenţia asupra pericolelor ce pîndesc tehnica şi cultura.

Istoria se grăbeşte! Dacă la începutul lui 1989 erau cunoscuţi 30 de viruşi care puteau infecta un PC, după numai un an cercetătorii de la Universiatea Erasmus din Rotterdam cunoşteau deja peste 150 de viruşi care infectaseră peste 35 de milioane de PC-uri.

Există multe căi prin care un virus poate ajunge într-un PC: reţelele de calculatoare, schimbul de programe via telefon, dischetele străine, şi chiar programele profesionale, versiunile demonstrative, programele Public Domain, oricare din ele poate servi ca purtător unui dăunător electronic.

Dar ce sînt viruşii calculatoarelor ?

Un pericol cunoscut poate fi apreciat mai corect. Din acest motiv vă prezentăm mai întîi o scurtă caracterizare a viruşilor.

Principial viruşii sînt nişte programe mici, software ca şi oricare alt program. În aproape toate cazurile, un virus constă din trei părţi de program. Prima, şi cea mai importantă, controlează reproducerea. De regulă, viruşii se autoreproduc, înmulţindu-se deci fără ajutor din exterior. A doua parte de program, numită activator, eliberează în momentul în care sînt satisfăcute anumite condiţii - de exemplu o dată prestabilită - a treia parte a virusului, nucleul. Acest nucleu conţine scopul final al virusului, efectuarea anumitor operaţii.

Există şi viruşi care nu au nucleu, dar adesea în nucleu există o comandă de formatare a harddiskului. Organele de "reproducere" şi de "activare" transportă "scopul" şi camuflează virusul pentru a nu fi descoperit înainte de vreme.

Familiile de viruşi

Ca şi dăunătorii biologici şi viruşii pot fi clasificaţi pe familii şi grupe, determinant fiind tipul fişierelor şi programelor infectate.

"Viruşii sectorului de boot" se fixează pe dischete sau pe harddisk în zona din care, la pornire (boot), calculatorul citeşte anumite informaţii despre caracteristicile discului (numărul capetelor de scriere/citire, numărul de sectoare pe pistă, numărul total de sectoare, etc.). Dacă PC-ul a fost pornit odată de pe o dischetă sistem infectată (pornire, reset sau "Ctrl"/"Alt"/"Del"), virusul se aşează şi în sectorul de boot al harddiskului. De acolo va infesta aproape orice dischetă nou introdusă.

"Viruşii fişierelor COM" sînt viruşii care se agaţă de fişierele care au extensia ".COM" (de ex. COMMAND.COM). Virusul devine activ deîndată ce este lansat în execuţie un astfel de program care îl conţine. Această categorie de viruşi, adesea, mai au o caracteristică: sînt rezidenţi în memorie, deci se fixează în memoria calculatorului independent de "purtător". Acolo pot deveni activi şi atunci cînd sînt executate alte programe.

Pornirea virusului este condiţionată, cel mai adesea, de tastarea unei anumite combinaţii de taste. Din memoria principală infectează alte programe cu extensia ".com" deîndată ce acestea sînt activate.

Ser, vaccin, profilaxie

Cine doreşte să-şi protejeze calculatorul împotriva viruşilor, trebuie să ţină cont de următoarele sfaturi:

- întotdeauna să fie păstrată în rezervă o copie (back-up) a programelor şi datelor importante, copie care să fie lipsită de viruşi
- să existe o dischetă sistem absolut lipsită de viruşi
- niciodată să nu se lucreze cu programele de pe dischetele originale. Dacă trebuie să lucraţi neapărat cu acestea, atunci protejaţi-le la scriere.
- controlaţi colecţiile de date pentru a vedea dacă nu au fost infectate cu:
  - fişierele de comenzi prezentate care compară lungimile fişierelor
  - programe anti-virus preţioase din categoria shareware, cum ar fi SCAN (Mcafee), care recunoaşte peste 60 de viruşi, sau cu VIRUSCAN al lui Peter van der Landen, care caută "semnătura" viruşilor.
- dacă descoperiţi un virus, există puţine metode de vindecare, şi anume:
  - cu anumite utilitare (PC Tools) virusul poate fi şters manual, sau ...
  - poate fi şters întregul fişier (de aceea sînt necesare copiile de siguranţă)
  - ... sau pot fi utilizate programe care ştiu să "ucidă" viruşii. Acestea elimină infecţia, însă nu lucrează întotdeauna fără greşeală.
- dacă aţi descoperit un virus în sectorul de boot, atunci opriţi calculatorul pentru cel puţin cinci secunde şi porniţi-l apoi de pe o dischetă sistem, neinfectată, protejată la scriere (pe dischetă trebuie să existe programul "SYS"). Daţi apoi comanda: "**SYS A: C:**". Dacă sistemul de operare reacţionează cu mesajul "System transferred" înseamnă că sectorul de boot a fost recopiat de pe discheta originală. Cu această operaţie este şters şi virusul.
- pentru o supraveghere continuă a sistemului, casa americană de soft Mcafee oferă un alt produs shareware preţios: "SCANRES". Dacă acesta este încărcat în memorie imediat după pornirea sistemului (lansat de ex. din fişierul de comenzi AUTOEXEC), atunci el supraveghează sistemul şi de îndată ce apare o "semnătură" cunoscută încuie sistemul. Calculatorul trebuie pornit din nou de pe o dischetă sistem curată şi dăunătorul trebuie şters.

Dacă se porneşte calculatorul din nou, memoria va fi liberă de viruşi atît timp cît programele sistemului de operare (DOS), deci şi command.com, nu sînt infectate.

"**Viruşii fişierelor EXE**" sînt asemănători ca tip cu viruşii fişierelor COM, doar că infectează numai fişierele cu extensia ".EXE". Există viruşi care infectează atît fişierele cu extensia .COM cît şi pe cele cu extensia .EXE. Un exemplu din această categorie de viruşi este de exemplu virusul DATACRIME II.

Pe lîngă tipurile numite de viruşi, există şi variante atipice. Acestei categorii îi aparţin viruşii care atacă fişierele batch (cu extensia .bat).

Tot la fel de periculoşi sînt şi "**caii troieni**". Un cal troian este un program complet cu o funcţie fermecătoare (demo de grafică, minibază de date, etc.), care neobservat de utilizator mai rezolvă şi o altă sarcină. În timp ce utilizatorul utilizează programul fără grijă, acesta poate formata discul pe ascuns. Hoţii de date, cum este şi renumitul cal troian Hack de la NASA, au fost construiţi, adesea, pentru a pătrunde în calculatoare străine şi a fura informaţii importante, drepturi de acces şi parole.

La începutul lui ianuarie '90 a debutat pe scena calculatoarelor un cal troian foarte periculos. Programul sfătuia în problematica SIDA şi evalua riscul personal al utilizatorului. În culisele programului se ascundea însă un cal troian. El cifra informaţia de pe harddisk şi îl bloca după 90 de boot-uri. În mod impertinent era oferit imediat şi un program antidot, dar utilizatorul trebuia să depună 100 DM într-un cont din Panama pentru a-l primi prin poştă.

Posesorii de PC-uri trebuie să cerceteze cu atenţie conţinutul fişierului "AUTOEXEC.BAT", deoarece viruşii de acest gen (denumiţi "AIDS" sau "Panama") depun în acest fişier o linie cu conţinutul "REM .EXE". Spaţiul liber, neuzual, din specificatorul de fişier nu conţine un spaţiu (cod 32), ci conţine codul ASCII 255. Dacă găsiţi această linie în fişierul d-voastră, ştergeţi-o şi apoi daţi comanda: "DEL REM?.EXE". Dacă nu reuşiţi, s-ar putea întîmpla ca fişierul "REM .EXE" să aibă atributul hidden setat - deci să fie ascuns. Cu ajutorul comenzii DOS "ATTRIBUTE" fişierul poate fi făcut vizibil şi apoi poate fi şters.

O altă grupă de viruşi se numeşte "**viermi**". Scopul lor este: să se reproducă atît de des, sau să se atîrne de alte fişiere pînă cînd nu mai este loc pe harddisk. Viermii scriu uneori şi peste datele existente.

Dăunătorii la lucru

Programele virus pot face tot ceea ce poate face calculatorul, deci pot modifica date, pot şterge, pot formata sau bloca. Din fericire în calea viruşilor pot fi puse şi anumite stavile. Aşa este, de exemplu, protejarea dischetelor flexibile la scriere prin lipirea unei etichete de protecţie. Unităţile floppy verifică cu ajutorul unui palpator, sau cu ajutorul unei raze de lumină, dacă eticheta de protecţie este lipită sau nu. Dacă da, atunci pe dischetă nu se mai poate scrie, iar virusul nu se

Do it yourself

Peter van der Landen, cercetător la Universitatea Erasmus din Rotterdam, a adunat în tabela alăturată "semnăturile" celor mai răspîndiţi viruşi. Prima coloană a tabelului conţine numele virusului, în a doua coloană este indicat locul în care se ascunde virusul, iar coloana a treia conţine identificatorul virusului în hexazecimal. Dacă depistaţi una dintre "semnăturile" hexa într-un fişier, de exemplu cu ajutorul funcţiei "Find" a programului "PC-Tools", căruia va trebui să-i indicaţi manual "semnătura", atunci puteţi şterge respectiva semnătură şi puteţi spera că programul va funcţiona în continuare. În caz contrar va trebui să ştergeţi întregul fişier şi să-l restauraţi apoi de pe o dischetă ce conţine o copie de siguranţă.

poate reproduce pe disc şi nici nu-l poate formata. Această limită fizică nu poate fi depăşită de nici un virus cunoscut. Şi totuşi, la unele unităţi floppy mai noi, controller-ul care comandă unitatea este programabil. Unii viruşi folosesc acest truc şi pun unitatea la lucru. Deci nu există o siguranţă absolută, dar vă recomandăm totuşi să protejaţi la scriere dischetele importante.

Încă şi mai periculoşi sînt viruşii care şterg sau chiar formatează harddisk-ul. În al doilea caz datele nu mai pot fi restaurate nici cu ajutorul utilitarelor. Comenzile de ştergere, de exemplu pentru o pistă a harddisk-ului, au din păcate lungimea de numai cîţiva biţi, astfel încît pot fi strecurate cu uşurinţă într-un dăunător minuscul. În continuare vă prezentăm o astfel de secvenţă de comenzi în limbaj de asamblare, care poate fi foarte uşor găsită în orice manual DOS:

Faza de pregătire

```
MOV AH, 05
    /* Formatare pistă
MOV DL, 00
    /* Numărul unităţii floppy 00 = A:
MOV DH, 00
    /* Numărul capetelor de scriere/citire
MOV CH, 00
    /* Numărul pistei de formatat
MOV CL, 01
    /* Sectorul de formatat: 1
MOV AL, 08
    /* Numărul de sectoare: 8
```

Aşa pot fi recunoscuţi viruşii

Nume	Infectează	Semnătura
Pakistani Brain	BOOT	8CC88ED88ED0BC00F0FBA0067CA2097C8 B0E077C890E0A7CE85700
Stoned (Marihuana)	BOOT	1E5080FC02721780FC0473120AD2750E33C08 ED8A03F04A8017503E80700
Yale (Alameda)	BOOT	BB40008EDBA11300F7E32DE0078EC00E1F 81FF56347504FF0EF87D
Ping Pong	BOOT	8ED8A113042D0200A31304B106D3E02DC 0078EC0BE007C8BFEB90001
Den Zuk	BOOT	FA8CC88ED88ED0BC00F0FBB8787C50C3
17XX	COM	F6872A0101740F8DB74D01BC
1701	COM	FA8BECE800005B81EB31012EF6872A0101740F 8DB74D01BC820631343124464C75F8
1704 of "1704-B"	COM	FA8BECE800005B81EB31012EF6872A0101740F 8DB74D01BC8506313431244 64C75F8
17Y4	COM	FA8BCDE800005B81EB31012EF6872A0101740F 8DB74D01BC8506313431244 64C75F8
1704-C	COM	F6872A0101740F8DB74D01BC8506313431244 64C77F8
1 April EXE	EXE	2EA31700BB17000E1FB4DECD21B42ACD2181F A0104742281F9BC077506E8C504
1 April COM	COM	89263401B419CD2104412EA265032EA2B103BF 6703578BF2807C013A750D8A042E A265032E A2B103
Jerusalem A	BOOT, COM, EXE	2EFF0E1F00EB122EC7061F
Jerusalem B	COM, EXE	8ED0BC000750B8C50050CBFC062E8C06310 02E8C0639002E8C063D002E8C0641008CC0
Jerusalem B-2	BOOT, COM, EXE	E9920000000000000000001
648	COM	FC8BF281C60A00BF0001B90300F3A48BF2B 430CD213C007503E9C701
DATACRIME 1280	COM	8B36010183EE038BC63D00007503E90201
DATACRIME 1168	COM	8B36010183EE038BC63D00007503E9FE00
DATACRIME II	COM, EXE	5E81EE030183FE00742A2E8A94
Lehigh	COM	505380FC4B740880FC4E7403E977018BDA80 7F013A75058A07EB07
405	COM, EXE	B8000026A2490226A24B0226A28B0250B419C D2126A24902B4470401
3066	COM, EXE	E87106E82806B419CD2189B45101818451018 4088C8C5301
Palet	COM	EB2B905A45CD602EC606250601902E803E26 06008D3E08060E07755E2EC6062 6
1701/1704 B	BOOT, COM, EXE	31343124464C75F8
1701/1704 C	BOOT, COM, EXE	31343124464C77F8
1168	BOOT, COM, EXE	EB00B40ECD21B4
Vacsina	BOOT, COM, EXE	56414353494E41
Golden Gate C	BOOT	A717800FADBDAD5507173384
Fu Manchu A	BOOT, COM, EXE	72454D484F72
Golden Gate C2	BOOT	A717DDAFF001233907173385
Nichols	BOOT	DD0DDF0FDD0FFF0AAA0ABA00
2730	BOOT	9177917AA4B7570056000000
Vienna (DOS 62) B	BOOT, COM, EXE	8BFE83C71F908BDE83C61F90
Vienna (DOS 62) A	BOOT, COM, EXE	8BFE81C71F008BDE81C61F00
1 April C	BOOT, COM, EXE	73555249560031
Jerusalem D	BOOT, COM, EXE	73555249560032
JerusalemE	BOOT, COM, EXE	73555249560033 3066
(Traceback)	BOOT, COM, EXE	E87106E82806B419CD21
Pentagon virus	BOOT	EB349048414C2020
Saratoga/Icelanding	BOOT, COM, EXE	A3030003D8438EC333F633FF
Icelandic B	BOOT, COM, EXE	2E8E1E6D02B90030DE0000
Israeli Boot	BOOT	CD13B80202B90627BA0001
Typo Boot	BOOT	241355AA
Ohio	BOOT	EB2990493412000100000000
3555	BOOT, COM, EXE	3306140031044646E2F3

Faza activă

```
INT 13
   /* Declanşarea întreruperii BIOS
   /* pentru accesul la dischetă
MOV AH, 00
   /* Pregătirea sfîrşitului programului
INT 21
   /* Programul este terminat normal
```

Din păcate pentru a putea identifica astfel de succesiuni de comenzi, este necesară o unealtă care să dezasambleze codul programului. Înafară de aceasta, comenzile de formatare pot fi camuflate astfel încît să nu poată fi recunoscute.

Partea vizibilă a aisbergului

Vă prezentăm în continuare cartea de vizită a cîtorva viruşi.

Cascade - mai întîi o literă, apoi tot mai multe cad pe ecran stricînd textul afişat, făcînd astfel lucrul imposibil.

Jerusalem - nu devine activ atîta timp cît data de 13 nu cade pe o zi de vineri, conform datei din memoria calculatorului indicată la încărcarea sistemului de operare. Dacă data de 13 este o zi de vineri şi virusul este activ, atunci se şterg, pur şi simplu, programele lansate.

Dark Avenger - se integrează în interiorul fişierelor, la fiecare 512 bytes, distrugîndu-le practic. Prezenţa virusului cauzează frecvente opriri şi îngheţări ale sistemului.

Sylvia/Holland - adaugă la sfîrşitul fişierelor numele şi numărul de telefon ale unei fete numite Sylvia, din Olanda.

Sunday - ceartă utilizatorii care lucrează duminica, afişînd mesajul: "Today is Sunday, why do you work so hard ?".

Yankee Doodle - în fiecare după-masă la ora 17 cîntă melodia "Yankee Doodle Dandy"

Cum putem lupta împotriva viruşilor ?

Înainte de a vaccina un calculator împotriva viruşilor, trebuie stabilit mai întîi dacă este "sănătos". După cum poate fi observat din tabelul alăturat, cel mai adesea, viruşii modifică lungimea originală a fişierelor. Acest lucru este de înţeles, pentru că ei se ataşează de programe fără să le deranjeze în funcţionare.

Viruşii pot fi depistaţi deci prin compararea lungimii fişierelor. Comanda DIR însă, care afişează şi lungimea fişierelor, nu rezolvă însă întotdeauna acest lucru, deoarece valoarea afişată a lungimii fişierului poate fi manipulată şi rescrisă de virus cu vechea valoare, normală. Totuşi nu orice virus foloseşte acest principiu de camuflaj.

Controalele simple, cum ar fi compararea lungimii fişierelor, sînt efectuate de programele anti-virus uzuale. Supărător este însă faptul că unele dintre aceste programe sînt oferite ca panaceu universal. Acest control poate fi efectuat foarte simplu şi de orice utilizator. Pentru aceasta este suficientă comanda: "DIR LPT1:", care trebuie repetată pentru fiecare subdirector, obţinîndu-se astfel lista cu lungimile originale ale fişierelor, care poate fi ulterior comparată cu lista lungimilor reale. Cu puţină osteneală, poate fi scris un fişier de comenzi care să producă automat o listă cu toate directoarele declarate în variabila PATH (în fişierul de comenzi AUTOEXEC.BAT). Fişierul "ARATA.BAT" apelează automat fişierul de comenzi "ARATA1.BAT", pentru fiecare director descoperit.

BATCH: **ARATA.BAT**

```
@echo off
for %%v in (%path%) do call arata1 %%v
type temp | more
rem del temp
```

BATCH: **ARATA1.BAT**

```
dir %1 | find/v "<DIR>" >> temp
```

Pentru alte verificări, poate fi scris un alt fişier de comenzi (batch), care să scrie într-un fişier lungimile tuturor fişierelor care au extensia .COM, .EXE sau .SYS. După o primă rulare fişierul "TEMP" trebuie redenumit (cu comanda DOS RENAME) în **"DOS_COMP.INI"** şi abia ulterior din ultima linie a fişierului de comenzi "DOS_COMP.BAT" trebuie şters cuvîntul "REM". În timpul rulărilor ulterioare, fişierul de comenzi va compara lungimea actuală a fişierelor cu lungimea originală.

BATCH: **DOS_COMP.BAT**

```
@echo off
dir c:\dos | find " EXE"  > temp
dir c:\dos | find " COM" >> temp
dir c:     | find " COM" >> temp
dir c:     | find " SYS" >> temp
comp dos_comp.ini temp
rem del temp
```

Practica dovedeşte însă că metodele care compară lungimea lasă suficiente găuri prin care se pot strecura viruşi. Pe de o parte unii viruşi nu modifică lungimea vizibilă a fişierelor, iar pe de altă parte există fişiere, cum ar fi fişiere de text, colecţii de date, biblioteci, care îşi schimbă natural lungimea. În aceste situaţii trebuie găsite alte metode de căutare.

Cel mai mare succes îl promit metodele care caută aşa-zisa "semnătură" a viruşilor: un fel de carte de vizită pe care viruşii o introduc într-un fişier pentru a nu-l infecta de mai multe ori. Vă prezentăm şi un tabel cu "semnăturile" celor mai cunoscuţi viruşi.

(Acest articol are la bază un articol publicat de Ralf Müller în revista Computer Live 3/90 şi a fost completat cu informaţii conţinute în documentaţiile programelor anti-virus PRGDOKI şi SCAN)

Lista principalilor viruşi cunoscuţi

(Copyright 1989, McAfee Associates)

Următoarea listă scoate în evidenţă caracteristicile critice ale principalilor viruşi cunoscuţi pe calculatoarele compatibile IBM PC.

Virus	Dezinfectant	Infectează							Rămîne rezident în memorie	Se autocriptează	Increment dimensiune program	Afectează
		tabelă partiţii harddisk	sector boot harddisk	sector boot dischetă	fişierele overlay	fişierele EXE	fişierele COM	fişierul COMMAND.COM				
V2000	SCAN/D	-	x	x	x	x	x	-		-	-	2000O,P,L
1554	SCAN/D	-	x	x	x	x	-	-	-	-	1554	O,P,L
512	SCAN/D	-	x	x	x	-	-	-	-	-	fără	O,P,L
EDV	M-DISK	-	x	-	-	-	-	x	x	x	N/A	B,O
Joker	CleanUp	-	x	x	x	-	-	-	-	-		O,P
Icelandic-3	CleanUp	-	x	-	-	x	-	-	-	-	853	O,P
Virus-101	CleanUp	x	x	x	x	x	x	x	-	-	2560	P
1260	CleanUp	x	-	-	x	-	-	-	-	-	1260	P
Perfume	CleanUp	-	-	-	x	-	-	-	-	-	765	P
Taiwan	CleanUp	-	-	-	x	-	-	-	-	-	708	P
Chaos	MDISK	-	x	-	-	-	-	x	x	-	N/A	B,O,D,F
Virus-90	CleanUp	-	x	-	x	-	-	-	-	-	857	P
Oropax	CleanUp	-	x	-	x	-	-	-	-	-	2773	P,O
4096	CleanUp	-	x	x	x	x	x	-	-	-	4096	D,O,P,L
Devil's Dance	CleanUp	-	x	-	x	-	-	-	-	-	941	D,O,P,L
Amstrad	CleanUp	-	-	-	x	-	-	-	-	-	847	P
Payday	CleanUp	-	x	-	x	x	x	-	-	-	1808	P
Datacrime II-B	CleanUp	x	-	x	x	x	-	-	-	-	1917	P,F
Sylvia/Holland	CleanUp	-	-	-	x	-	-	-	-	-	1332	p
Do-Nothing	CleanUp	-	x	-	x	-	-	-	-	-	608	p
Sunday	CleanUp	-	x	-	x	x	x	-	-	-	1636	O,P
Lisbon	CleanUp	-	-	-	x	-	-	-	-	-	648	P
Typo/Fumble	CleanUp	-	x	-	x	-	-	-	-	-	867	O,P
Dbase	CleanUp	-	x	-	x	-	-	-	-	-	1864	D,O,P
Ghost Boot Version	MDISK	-	x	-	-	-	-	x	x	-	N/A	B,O
Ghost COM Version	CleanUp	-	-	-	x	-	-	-	-	-	2351	B,P
New Jerusalem	CleanUp	-	x	-	x	x	x	-	-	-	1808	O,P
Alabama	CleanUp	-	x	-	-	x	-	-	-	-	1560	O,P,L
Yankee Doodle	CleanUp	-	x	-	x	x	-	-	-	-	2885	O,P
2930	CleanUp	-	x	-	x	x	-	-	-	-	2930	P
Ashar	CleanUp	-	x	-	-	-	-	x	-	-	N/A	B
AIDS	CleanUp	-	-	-	x	-	-	-	-	-	suprascrie programul	
Disk Killer	CleanUp	-	x	-	-	-	-	x	x	-	N/A	B,O,P,D,F
1536/Zero Bug	CleanUp	-	x	-	x	-	-	-	-	-	1536	O,P
MIX1	CleanUp	-	x	-	-	x	-	-	-	-	1618	O,P
Dark Avenger	CleanUp	-	x	x	x	x	x	-	-	-	1800	O,P,L

3551/Syslock	CleanUp	x	-	-	-	x	x	-	-	-	-	-	3551	P,D
VACSINA	CleanUp	-	x	-	-	x	x	x	-	-	-	-	1206	O,P
Ohio	MDISK	-	x	-	-	-	-	-	x	-	-	-	N/A	B
Typo (Boot Virus)	MDISK	-	x	-	-	-	-	-	x	x	-	-	N/A	O,B
Swap/Israeli Boot	MDISK	-	x	-	-	-	-	-	x	-	-	-	N/A	B
1514/Datacrime II	CleanUp	x	-	-	-	x	x	-	-	-	-	-	1514	P,F
Icelandic II	CleanUp	-	x	-	-	-	x	-	-	-	-	-	661	O,P
Pentagon	MDISK	-	-	-	-	-	-	-	x	-	-	-	N/A	B
3066/Traceback	M-3066	-	x	-	-	x	x	-	-	-	-	-	3066	P
1168/Datacrime-B	CleanUp	x	-	-	-	x	-	-	-	-	-	-	1168	P,F
Icelandic	CleanUp	-	x	-	-	-	x	-	-	-	-	-	642	O,P
Saratoga	CleanUp	-	x	-	-	-	x	-	-	-	-	-	632	O,P
405	CleanUp	-	-	-	-	x	-	-	-	-	-	-	suprascrie programul	
1704 Format	CleanUp	x	x	-	-	x	-	-	-	-	-	-	1704	O,P,F
Fu Manchu	CleanUp	-	x	-	-	x	x	x	-	-	-	-	2086	O,P
1280/Datacrime	CleanUp	x	-	-	-	x	-	-	-	-	-	-	1280	P,F
1701/Cascade	CleanUp	x	x	-	-	x	-	-	-	-	-	-	1701	O,P
1704/CASCADE-B	CleanUp	x	x	-	-	x	-	-	-	-	-	-	1704	O,P
Stoned/Marijuana	CleanUp	-	x	-	-	-	-	-	x	-	x	-	N/A	O,B,L
1704/CASCADE	CleanUp	x	x	-	-	x	-	-	-	-	-	-	1704	O,P
Ping Pong-B	CleanUp	-	x	-	-	-	-	-	x	x	-	-	N/A	O,B
Den Zuk	MDISK	-	x	-	-	-	-	-	x	-	-	-	N/A	O,B
Ping Pong	CleanUp	-	x	-	-	-	-	-	x	-	-	-	N/A	O,B
Vienna-B	CleanUp	-	-	-	-	x	-	-	-	-	-	-	648	P
Lehigh	CleanUp	-	x	x	-	-	-	-	-	-	-	-	suprascrie	P,F
Vienna/648	M-VIENNA	-	-	-	-	x	-	-	-	-	-	-	648	P
Jerusalem-B	CleanUp	-	x	-	-	x	x	x	-	-	-	-	1808	O,P
Yale/Alameda	CleanUp	-	x	-	-	-	-	-	x	-	-	-	N/A	B
Friday 13th COM	CleanUp	-	-	-	-	x	-	-	-	-	-	-	512	P
Jerusalem	CleanUp	-	x	-	-	x	x	x	-	-	-	-	1808	O,P
SURIV03	CleanUp	-	x	-	-	x	x	x	-	-	-	-		O,P
SURIV02	CleanUp	-	x	-	-	-	x	-	-	-	-	-	1488	O,P
SURIV01	CleanUp	-	x	-	-	x	-	-	-	-	-	-	897	O,P
Pakistani Brain	CleanUp	-	x	-	-	-	-	-	x	-	-	-	N/A	B

Caracteristici x - da - nu

Dezinfectant - programul care poate elimina infecţia
SCAN/D - VIRUSCAN cu opţiunea /D
SCAN/D/A - VIRUSCAN cu opţiunile /D şi /A
MDISK/P - MDISK cu opţiunea "P"
toate celelalte - numele programului de dezinfectare

Legendă:
Cîmpuri alterate
B - corupe sau suprascrie sectorul de boot
O - alterează sistemul de operare
P - corupe programele sau fişierele overlay
D - corupe fişierele de date
F - formatează sau şterge în întregime/parţial discul
L - direct sau indirect corupe fişiere individuale

Incrementul dimensiunii programului - lungimea în bytes, cu care este incrementat un program sau un fişier overlay infectat

Notă:
Opţiunea SCAN /D va rescrie şi apoi va şterge întregul program infectat. Programul trebuie retransferat apoi de pe discheta originală. Dacă doriţi să încercaţi să recuperaţi un program infectat, atunci utilizaţi programul dezinfectant specificat (dacă îl aveţi).

O salvare corectă

Salvările (backup-urile) sînt importante, ele sînt folositoare însă numai atunci cînd şi strategia de salvare este corespunzătoare. În această privinţă vă putem da cîteva sugestii.

Cine îşi cumpără un PC cu harddisk, va găsi, aproape întotdeauna, în manualul de utilizare indicaţia de a-şi crea o copie de siguranţă a datelor. Această copie este necesară pentru situaţia în care datele de pe harddisk se pierd datorită unei erori de operare sau datorită unui ghinion. Dar pentru ca o astfel de copie să fie utilizabilă trebuie ca ea să fie actuală şi să conţină toate datele relevante.

Pentru a afla drumul cel bun trebuie consumat însă timp şi este necesară o oarecare doză de autodisciplină. Utilizatorii profesionişti ştiu desigur că după instalarea tuturor programelor importante pe harddisk este necesară o salvare a întregului disc, păstrîndu-se astfel o copie suplimentară a preţioaselor programe. Nu se recomandă însă ca această copie să fie o "copie imagine", 1:1 (mod sector), care să utilizeze toţi parametrii discului şi care să folosească toţi aceşti parametri şi la restaurare. Dacă în urma unei reformatări, de ex., se schimbă numărul sectoarelor defecte, atunci se modifică parametrii discului şi salvările efectuate cu parametrii vechi nu mai pot fi utilizate.

Înainte de salvarea unui harddisk acesta trebuie verificat cu programe anti-virus pentru a fi siguri că vom obţine o copie "curată". Chiar dacă nu ne putem încrede 100% în programele anti-virus, neexistînd protecţie perfectă, ne asigurăm în acest mod că măcar acei viruşi pe care programul îi cunoaşte nu vor ajunge în copia de siguranţă.

Pentru a avea o copie actuală a datelor, acestea trebuie salvate la intervale regulate de timp. Lungimea intervalului trebuie stabilită în funcţie de frecvenţa de modificare a datelor. Utilizatorii profesionali - din întreprinderi mari - nu-şi rezolvă de regulă problema decît făcînd mai multe salvări zilnic. Unele programe de salvare oferă în plus posibilitatea de a salva fişierele după dată sau după extensie. Este posibilă astfel salvarea doar a fişierelor care au fost create după 15 februarie, de ex., sau doar a fişierelor cu terminaţia "*.DOC".

În orice caz, după fiecare modificare semnificativă a colecţiei de date ar trebui efectuată o salvare. În aceste cazuri este indicat să se folosească două suporturi de memorare diferite, care să fie utilizate alternativ. Astfel salvările impare (prima, a treia, etc.) ar urma să fie salvate pe un suport, iar cele pare (a doua, a patra, etc.) pe celălalt suport. Dacă se lucrează în acest mod, chiar dacă una din copii devine inutilizabilă mai rămîne o copie de siguranţă, economisindu-se astfel timp de lucru preţios.

În cazul utilizatorilor care utilizează PC-ul la birou, aproape că nu mai începe nici o discuţie, la sfîrşitul programului de lucru, salvările, dischetele şi celelalte medii de memorare amovibile, sînt încuiate pentru a fi protejate într-un seif special. Acolo este locul ideal pentru aceste medii, deoarece aceste "magazii" sînt reci, întunecoase şi protejate împotriva eventualelor cîmpuri magnetice.

Dacă însă nu dispunem de astfel de locuri de depozitare, trebuie să avem grijă unde ţinem copiile de siguranţă, deoarece căldura prea mare, variaţiile semnificative de temperatură sau cîmpurile magnetice pot altera datele. Din acest motiv nu este indicată stocarea acestor medii în apropierea monitoarelor sau televizoarelor.

Mai există încă un factor care nu trebuie scăpat din vedere: nici un mediu de memorare nu este veşnic. Benzile casetelor streamer-

elor, de ex., ar trebui schimbate cel puţin odată pe an, deoarece pelicula lor magnetică se uzează în timp şi pot apare pierderi de date (drop-outs).

Plecă un tren spre nicăieri ...

Nenumărate lacrimi au vărsat deja utilizatorii de calculatoare ale căror date au pornit într-o călătorie spre nicăieri. Lor nu le mai pot fi de folos decît salvările actuale. Care mediu de memorare este însă cel mai indicat pentru asigurarea datelor ?

Cîntăreţii de şlagăre se vaită din toate băierile inimii atunci cînd i-a părăsit iubita. Dar vaietele lor încă nu sînt nimic pe lîngă cele ale utilizatorilor de calculatoare, atunci cînd trăiesc dezastrul unei dispariţii totale a datelor - preţioşii biţi dispărînd în nirvana unei dischete sau a unui harddisk.

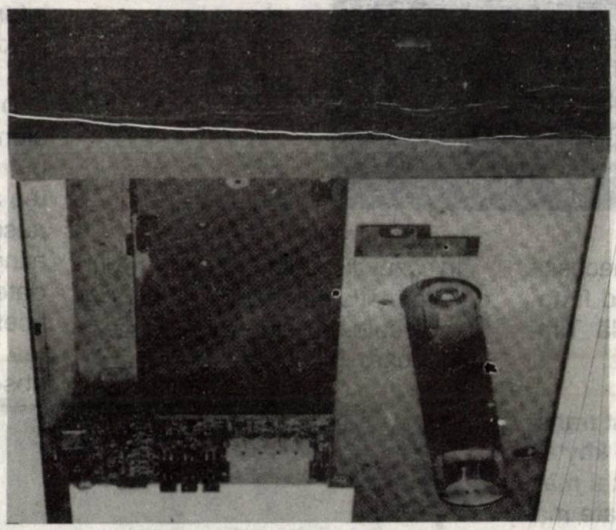
După o astfel de "bucurie" se scotoceşte după dischetele care ar trebui să conţină o salvare de pe harddisk, se dă comanda "Restore", şi urmează marea dezamăgire: nu mai există nici o urmă a datelor, salvarea este îmbătrînită şi nu mai este nici o speranţă.

Din aceste motive sînt căutate medii de salvare comode. Capacitatea mare de memorare singură, nu este suficientă însă, o fiabilitate ridicată este cel puţin tot la fel de importantă. Aceste considerente ne-au determinat să vă prezentăm în continuare un streamer: Wangtek 5150 (cca. 1860 DM), un disc magneto-optic: Megapower AT produs de K&S (cca. 11300 DM) şi un harddisk amovibil: Bernoulli-Box 44e produs de Iomega (cca. 3700 DM).

Pentru care scop este recomandat fiecare sistem ? Satisfac aceste aparate şi cerinţele de fiabilitate şi de utilizare confortabilă ?

Wangtek 5150

Streamer-ul Wangtek 5150 este construit compact şi în mod standard se montează în interiorul PC-



ului. La cerere poate fi livrat însă şi pentru uz extern.

Furnitura de livrare conţine o plachetă SCSI, produsă de Adaptec, care dispune de un controller propriu de dischetă, şi deci nu trebuie cumpărată o plachetă separată.

Streamer-ul este livrat împreună cu software-ul Sytos. La instalarea acestuia trebuie modificat fişierul "CONFIG.SYS", la începutul acestuia trebuind introdusă linia: "DEVICE = ASPI4DOS.SYS". Programul permite salvarea totală "imagine" sau "fişier" a harddisk-ului, sau salvarea doar a unor fişiere specificate (după extensie, director sau dată). Şi restaurarea poate fi făcută selectiv.

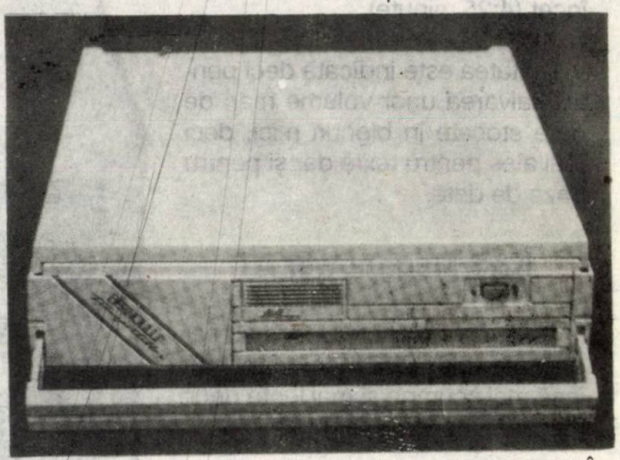
Rata de transfer este foarte ridicată: mai mult de 31 Mbytes (stocaţi în 400 de fişiere de 18 kbytes şi şapte fişiere de 3 Mbytes) au fost transferaţi pe bandă, în timpul unui test, în 3:09 minute, deci se atinge o rată de transfer de 176 kbytes /secundă.

Streamer-ul este recomandabil deci pentru utilizatorii pentru care salvările perfecte sînt foarte preţioase şi care nu vor să piardă timp preţios efectuîndu-le.

Iomega Bernoulli 44e

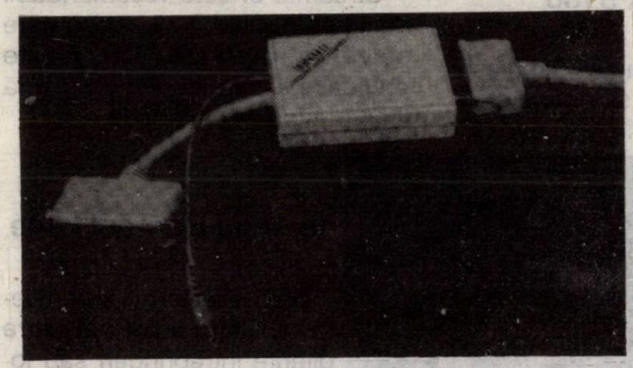
Necesitatea de a transfera cantităţi mari de date între diferite întreprinderi sau locuri de muncă, nu poate fi satisfăcută corespunzător doar cu dischete, care pot memora doar cca. 1 Mbyte. În aceste situaţii confortul dorit nu poate fi obţinut decît cu harddisk-uri amovibile.

Iomega Bernoulli 44e este o unitate pentru harddisk-uri



amovibile, care costă 3700 DM. În preţ nu este inclus şi preţul harddisk-ului propriu-zis (o bucată 390 DM, un pachet de 3 bucăţi 1100 DM). Odată cu unitatea se livrează şi un driver specific şi un program de instalare care copiază în sistem fişierele necesare şi modifică fişierele CONFIG.SYS şi AUTOEXEC.BAT.

Programul de salvare permite fie o copiere exhaustivă a harddisk-ului fie o copiere selectivă. În pachet este conţinut şi un program de formatare, care permite stabilirea diferiţilor parametrii ai harddisk-ului amovibil, cum ar fi mărimea partiţiilor sau factorul interleave. Cu ajutorul lui un disc de 44 Mbytes poate fi partiţionat în două sau mai multe unităţi logice.



Cablul special poate fi legat la interfaţa paralelă

Unitatea Bernoulli nu necesită un controller special putînd fi conectată pe interfaţa paralelă. Timpul de acces este de 28 ms.

În timpul unui test au fost transferate 400 de fişiere de 18 kbytes în 3:57 minute, blocurile mai mari (3 Mbytes) fiind transferate mai încet (4:25 minute).

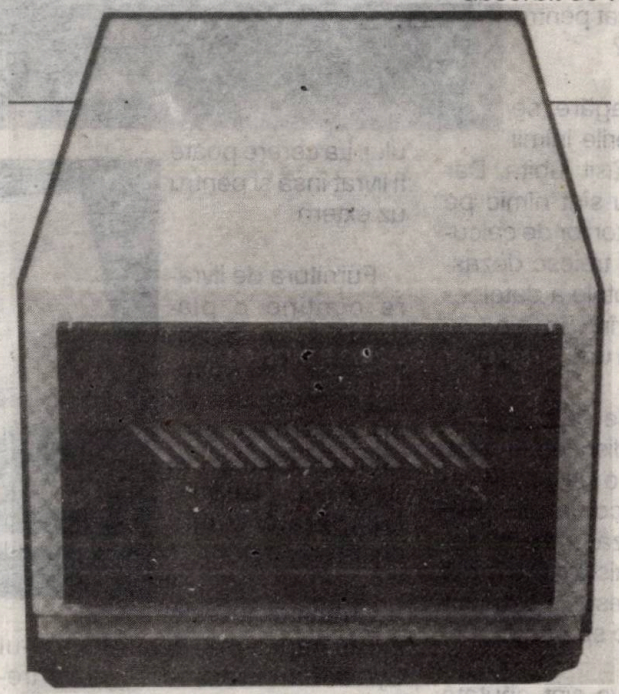
Unitatea este indicată deci pentru salvarea unor volume mari de date stocate în blocuri mici, deci mai ales pentru texte dar şi pentru baze de date.

K&S Megapower AT

Discurile magneto-optice au capacitate uriaşă de memorare, sînt însă mai scumpe şi mai lente decît mediile anterior prezentate.

K&S Megapower AT este o unitate magneto-optică destul de scumpă. Unitatea costă 11.300 DM, iar un disc magneto-optic costă 1000 DM şi are o capacitate de memorare de aproape 600 Mbytes.

În cazul discurilor magneto-optice informaţia este "arsă" pe suprafaţa discului cu ajutorul unei raze laser, care încălzeşte pelicula la



exact 160 grade Celsius (deci nici 159 şi nici 161), temperatură la care pelicula îşi modifică caracteristicile magnetice.

Cîmpul magnetic modificat schimbă polarizarea razei laser la citirea discului, schimbare înregistrată de un fotoelement care recepţionează lumina. În acest mod informaţiile pot fi scrise şi citite de pe disc.

Dacă pe suprafaţa discului ajung însă particule de praf, fie şi numai de cîţiva micrometri, acestea absorb pentru un moment minuscul raza laser, pelicula nu mai atinge temperatura planificată, şi datele vor fi alterate. Din cauza acestui motiv K&S şi-a dotat unitatea cu un filtru care curăţă aerul absorbit de ventilator.

Unitatea este livrată împreună cu un software specific (driver, program de salvare/restaurare, etc.). În timpul testului au fost salvate 400 de fişiere de 18 kbytes şi şapte fişiere de 3 Mbytes în 15:57 minute. Fişierele mari au fost salvate mai uşor decît cele scurte.

Unitatea se recomandă a fi utilizată în cazul salvării unor volume mari de date (arhivări). Să sperăm că în viitor va creşte şi viteza de transfer a datelor.

(Computer Persönlich 26/90, Stephan Mayer)

Tabel comparativ

Nume	K&S Megapower AT	Iomega Bernoulli 44e	Wangteck 5150
Preţ aprox. (DM)	11.300	3.700	1.860
Intern/extern	extern	extern	intern
Capacitate	594 MBytes	44 MBytes	150 / 250 MBytes
Interfaţă	SCSI	paralelă	SCSI
Controller	Adaptec 1542	-	Adaptec 1542
Software de comandă	K&S - MODRIVER	RCD	Sytos

O ghilotină pentru fişiere

Cu ajutorul programului "**cutfile.c**" puteţi "tăia" în bucăţi un fişier care este prea mare şi nu încape, de ex., pe dischete de 360 Kbytes. Aceste bucăţi pot fi salvate pe dischete de capacitate mai mică şi după o restaurare pe un alt harddisk fişierul original poate fi refăcut din aceste bucăţi cu ajutorul comenzii "**copy**". Un alt exemplu de utilizare a lui "cutfile" ar fi împărţirea fişierelor care trebuie încărcate într-o cutie poştală (mailbox), în cazul în care trebuie respectată o lungime maximă a pachetului.

Programul a fost scris pentru compilatorul **Turbo C 2.0** şi trebuie linkeditat împreună cu modulul obiect "**Getopt**". Se va alege modelul de memorie "**small**". Dacă aţi obţinut imaginea executabilă a programului, atunci puteţi încerca un prim test dînd comanda:

cutfile NumeFişier </SMărimeBucată> </NNumărBucăţi> </ONumeFişierIeşire>

NumeFişier este identificatorul fişierului care urmează să fie împărţit în bucăţi. Dacă se doreşte ca bucăţile rezultate să aibă o anumită lungime, atunci această valoare poate fi dată ca parametru după opţiunea "**/S**". O altă modalitate este specificarea numărului de bucăţi (aproximativ egale), în care urmează a fi împărţit fişierul, cu ajutorul opţiunii "**/N**". Cei doi parametrii, "/S" şi "/N", se exclud reciproc. Lungimea implicită a bucăţilor este de **320 Kbytes**, astfel încît ele pot fi salvate comod pe dischete de 360 Kbytes. Parametrul "**/O**" permite specificarea numelui fişierului de ieşire. Implicit numele fişierului ii rămîne acelaşi ca la intrare, iar extensia fişierului este formată dintr-o succesiune de numere. Specificatorii fişierelor pot varia deci de la "NumeFişier.001" pînă la "NumeFişier.999". Dacă există deja un fişier cu specificatorul generat, atunci nu se scrie peste el ci se fac maxim **256** de încercări de a găsi un specificator de fişier neutilizat.

Pentru refacerea întregului din bucăţi nu este necesar un utilitar special, fiind suficientă comanda DOS "**copy**". O comandă de refacere ar putea arăta în modul următor:

copy /B fişier.001 + fişier.002 + fişier.003 întreg.dat

"**/B**" este un parametru al comenzii "copy". Acest parametru trebuie specificat, atunci cînd la copierea sau concatenarea de fişiere, dorim să ne asigurăm că operaţia nu va fi întreruptă la apariţia primului marcaj de sfîrşit de fişier **<CTRL/Z>**.

```
/*
     cutfile.c - fragmenteaza un fisier in bucati
     parametru: numefisier
     optiuni:
          /Sxxx - determina marimea fragmentelor
               (implicit 10*BUF_SIZE)
          /Nnnn - determina numarul fragmentelor
               (implicit se calculeaza din marimea
               fisierului si marimea implicita pentru
               un fragment)
          /Ofisier - determina fisierul de iesire
     (dupa Matthias Kring, DOS International)
*/

/*** fisiere header ***/
#include <stdio.h>
#include <dos.h>
#include <io.h>
#include <dir.h>
#include <errno.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <sys\types.h>
#include <sys\stat.h>

/*** Definitii ***/
#define ERR          (-1)
#define OK  1
#define F_OK         0       /* definitie X/OPEN ptr. apel access() */
#define BUF_SIZE 32768       /* 32 kbyte */
#define SIZE_DEF (10*BUF_SIZE)
                             /* marimea implicita a fragmentelor;
                                ar trebui sa fie aleasa in asa fel
                                incit sa incapa lejer pe o discheta
                                normala (360 kbyte), eventual impreuna
                                cu un utilitar sau un fisier help */
#define MAXPART 999      /* numarul maxim de fragmente */
#define MINSIZE 1  /* dimensiune minima a unui fragment */

/* pentru indicator (optiunile S si N se exclud reciproc) */
#define SIZE 1
#define PART 2

/* pentru getopt() */
extern char * optarg;
extern       int optind;
/* tamponul de citire */
unsigned char buf  [(unsigned) BUF_SIZE];

/******************** programul principal ***************/

void  main (argc, argv)
int argc;
char *argv[];

{
     /* declaratii */
     int ii, jj, filextcount, try, flag;       /* diversi contori */
     long int cc, size, fs, part;  /* size = marimea unui fragment
                                      part = numarul de fragmente */
     unsigned rc, readitems, rest;          /* pentru citiri */
     int ifh, ofh;                          /* input/output file handle */
     char filprefix  [MAXFILE];
```

```
char    infilnam [MAXPATH],
        outfilnam [MAXPATH],
        path     [MAXPATH];
struct stat statbuf, *sbp;
char *options = "s:S:n:N:o:O:hH";
void mess();
void usage();
void error_exit();
/* initializari importante */
cc = part = 0L;
sbp = &statbuf;
flag = try = 0;
size = SIZE_DEF;
outfilnam[0] = '\0';
/* interpretarea parametrilor din linia de comanda */
while ((ii = getopt(argc, argv, options)) != EOF)
{
        switch (ii)
        {
                /* marimea fragmentelor */
                case 's':
                case 'S': if (flag == 0)
                          {
                          size = atol (optarg);
                          flag = SIZE;
                          }
                          break;
                /* numarul fragmentelor */
                case 'n':
                case 'N': if (flag == 0)
                          {
                          part = atoi (optarg);
                          flag = PART;
                          }
                          break;
                /* fisierul de iesire */
                case 'o':
                case 'O': sprintf (outfilnam,"%s", optarg);
                          break;
                /* help */
                case 'h':
                case 'H':
                case '?': usage ();
        }
}
if (argc - optind != 1)
        /* mai e permis un singur argument ramas */
{
        usage();
}
if (size <= 01)
{
        size = MINSIZE;
        printf ("Marimea minima a unui fragment este %ld.\n",
                MINSIZE);
}
if ((part <= 01 && flag == PART ) || part > MAXPART)
{
        part = MAXPART;
        printf ("Fragmentez in %d parti (maximul).\n",
                MAXPART);
}

/* verifica fisierul sursa */

sprintf (infilnam,"%s", argv[optind]);

/* determina marimea fisierului */
if (stat (infilnam, sbp) == ERR)
{
        error_exit (infilnam);
}
fs = statbuf.st_size;
/* calculeaza numarul fragmentelor,
   respectiv marimea lor    */
if (flag != PART || part == 0L)
{
        if (fs <= size) part = 1L;
        else
        {
                if (fs % size > 0) part = 1L;
                part = part + fs/size;
        }
}
else    /* marimea unui fragment e data */
{
        if (fs <= part)
        {
                size = 1L;
                part = fs;
        }
        else
        {
                size = fs/part;
                if (fs % part > 0) size += 1L;
        }
}
if (part == 1L || size == fs || part > MAXPART)
{
  printf ("nu se poate fragmenta un fisier de %ld byte
        in %ld fragmente de %ld byte!\n", fs, part, size);
  exit (1);
}
printf ("Fisierul sursa de %ld byte va fi fragmentat in %ld
        parti de (maxim) %ld bytes.\n", fs, part, size);
if ((ifh = open (infilnam, O_RDONLY | O_BINARY ))
                                        == ERR)
{
        error_exit (infilnam);
}
/* verifica fisierul (fisierele) destinatie */
if (outfilnam[0] == '\0')
{
        strcpy (outfilnam, infilnam);
}
/* determina prefixul fisierelor */
/* infilnam folosit ca tampon - numai e nevoie de el */
if (strrchr (outfilnam, '\\') != NULL)
        sprintf (infilnam, "%s", &strrchr(outfilnam, '\\')[1]);
else
        sprintf (infilnam, "%s", outfilnam);

/* restul dupa ultimul \; mai poate consta insa din filename.ext! */

ii = strlen(outfilnam);
jj = strlen(infilnam);
strncpy (path, outfilnam, ii-jj);
path [ii-jj] = '\0';
ii = strcspn (infilnam, ".");
strncpy (filprefix, infilnam, ii);
filprefix [ii] = '\0';

/* prefixul definitiv */

sprintf (outfilnam, "%s%s", path, filprefix);
if (part > 50)
        printf ("\n Verificarea de nume de fisiere de iesire
                utilizabile poate dura. Putintica rabdare... \n");
while (file_exist (outfilnam, part) == ERR)
{
        if (try == 0)
                printf ("Fisierul de iesire exista. Caut un alt
                        nume de fisier.\n");
        if (try > 0xff)
        {
          printf ("Abandon. Toate numele valide de
                fisier de la %.5s_00 la %.5s_FF sint
```

```
                eja in uz.\n",filprefix, filprefix);
            printf ("Reincercati dupa ce mai stergeti din ele.\n");
            exit (1);
        }
        /* nume nou */
        sprintf (infilnam, "%.5s_%02X", filprefix, try);
        sprintf (outfilnam, "%s%s", path, infilnam);
        try = try + 1;
}
if (try != 0)
{
        printf ("Pentru iesire se foloseste numele de
                fisier%s.\n", outfilnam);
        sprintf (filprefix, "%s", infilnam);
}
filextcount = 0;
/* marimea tamponului trebuie acordata la marimea
                                        fragmentului;
    cc = numarul buclelor de citire a cite readitems,
                                        apoi restul */
if (size <= BUF_SIZE)
{
        cc = 1L;
        rest = 0;
        readitems = (unsigned) size;
}
else
{
        cc = size / BUF_SIZE;
        rest = (unsigned) (size % BUF_SIZE);
        readitems = (unsigned) BUF_SIZE;
}
/* citire si scriere */
while (eof(ifh) == 0 && part > 0)
{
        sprintf (outfilnam, "%s%s.%03d", path, filprefix,
                filextcount += 1);
        ofh = file_open (outfilnam);
        for (ii = 1; ii <= cc; ii++)
        {
                /* citeste numar readitems */
                if ((rc = read(ifh, &buf, readitems)) <= 0)
                {
                        if (eof (ifh) == 1)
                        {
                                buf[rc] = '\0';
                                break;
                        }
                        else if (rc == (unsigned) ERR)
                        error_exit ("read input");
                }
                if ((rc = write (ofh, buf, rc )) ==
                                        (unsigned) ERR)
                {
                        error_exit (outfilnam);
                }
        }
        if (rest > 0 && eof (ifh) == 0)
        {
                if ((rc = read (ifh, &buf, rest)) <= 0)
                {
                        if (eof (ifh) == 1)
                                buf[rc] = '\0';
                        else if (rc == (unsigned) ERR)
                        error_exit ("read input");
                }
                if ((rc = write (ofh, buf, rc )) ==
                                        (unsigned) ERR)
                {
                        error_exit (outfilnam);
                }
        }
        close (ofh); /* s-a terminat un fragment */
                part--;
    }
    printf ("Terminat.\n");
    mess (filprefix, filextcount);
    exit (0);
}
/********** sfirsitul programului principal **************/
/*  verifica toate numele de fisiere de iesire ; exista ? */
file_exist (filepath, parts)
char * filepath;
long int parts;
{
    char filename[MAXPATH];
    int ii;
    for (ii = 1; ii <= parts; ii++)
    {
            sprintf (filename, "%s.%03d", filepath, ii);
            if (access (filename, F_OK) == NULL)
            {
                    /* fisierul exista deja */
                    return (ERR);
            }
    }
    return (OK);
}
/* deschiderea unui fisier */
file_open (filename)
char * filename;
{
    int ofh;
    if ((ofh = open (filename, O_RDWR | O_BINARY |
                    O_CREAT | O_TRUNC, S_IREAD |
                    S_IWRITE )) == NULL)
    {
            error_exit (filename);
    }
    else
    {
            printf ("Scriu in fisierul '%s'\n",filename);
    }
    return (ofh);
}
/* utilizare */
void usage()
{
    printf ("\nIf Freeware (dupa Matthias Kring, pentru
                        DOS International).\n");
    printf ("\nUtilizare: Cutfile [/Sxxx | /Nnnn]
            [O <fisierdeiesire>] <fisiersursa>\n\n"
            "Acest utilitar fragmenteaza un fisier in bucati de cel
              mult xxx byte\n"
              sau in nnn parti de marimi (aproape) identice.\n"
            "Marimea implicita a unui fragment este %ld bytes.\n"
            "Fragmentele sint numite fisierdeiesire.xxx, unde xxx
                    este un numar de secventa.\n"
            , SIZE_DEF);
    mess ("outfile", 4);
    exit (1);
}
void mess (fn, n)
char * fn;
int n;
{
    printf ("Fisierul original poate fi refacut cu comanda DOS:\n"
        "copy /b %s.001 + ... + %s.%03d %s.org\n", fn,fn, n, fn);
}
void error_exit (name)
char * name;
{
    perror (name);
    exit (1);
}
```

Scrierea unui număr în litere

Programul **NUMAR.PRG**, scris în **FOX-BASE+**, are drept scop convertirea unui număr arbitrar cuprins între 0 şi 9999999999.99 în litere. În acest scop s-au definit şi folosit două fişiere:

A) **UNIT.DBF** cu structura:

	DEN.CIMP	TIP	DIMENSIUNE
1.	NUMAR	Numeric	2
2.	NUMAR_LIT	Caracter	13

Cîmpul 1. cuprinde numerele de la 0 la 19, iar cîmpul 2. cuprinde scrierea acestor numere în litere.

B) **NUMAR.DBF** cu structura:

	DEN.CIMP	TIP	DIMENSIUNE
1.	NUMAR	Numeric	1
2.	NUMAR_LIT	Caracter	8
3.	NUMAR_LI	Caracter	7

Fişierul NUMAR.DBF are următorul conţinut:

NUMAR	NUMAR_LIT	NUMAR_LI
0	miliarde	
1	milioane	sute
2	milioane	zeci si
3	milioane	
4	mii	sute
5	mii	zeci si
6	mii	
7	sute	
8	zeci si	

Zecimalele, daca sînt, vor fi scrise sub forma: 99/100. Indexarea celor 2 fişiere s-a făcut după cîmpurile "NUMAR".

(fiz. Radovici Vera)

```
* Program:  NUMAR.PRG
* Autor:    Radovici Vera
*
* Scopul programului:
* -transcrierea in litere a unui numar arbitrar
*
* Soft-ul folosit: FOX-BASE+ vers. 2.00
*
* se inchid toate fisierele deschise si se sterge
*ecranul
clos all
clea
* se stabileste contextul de lucru
set talk off
* se introduce numarul si i se stabileste ordinul
* de marime
aa=0
'9999999999.99' func 'z'
read
nr=13-len(ltrim(str(aa,13,2)))
nrn=10-nr
nu=nrn
ord=nr
* se deschid cele 2 fisiere care dau:
* -UNIT.DBF:scrierea in litere a numerelor de la
* 0 la 19
* -NUMAR.DBF:scrierea in litere a ordinelor de ma-
* rime (miliarde,milioane,mii,sute,zeci)
sele 1
use unit inde unit
sele 2
use numar inde numar
* urmatoarele 2 bucle fac corespondenta dintre fi-
* ecare cifra si scrierea ei si a ordinului ei de
* marime in litere
do whil nrn#0
    n=int(val(substr(ltrim(str(aa,13,2)),11-nr-nrn,;
1)))
    sele 1
    seek n
    or='n'+ltrim(str(nrn))
    &or=iif(n#2,rtrim(numar_lit),iif(nrn#1,'doua',;
rtrim(numar_lit)))
    sele 2
    seek ord
    or1='o'+ltrim(str(nrn))
    if (ord#8.or.(ord=8.and.n#1)).and.(ord#2.or.(o;
rd=2.and.n#1)).and.(ord#5.or.(ord=5.and.n#1))
     &or1=rtrim(numar_li)+' '+rtrim(iif(nrn=10.o;
r.nrn=7.or.nrn=4.or.nrn=1.or.nrn=3.or.nrn=2;
,numar_lit,''))
        ord=ord+1
    else
        nrn=nrn-1
        n=int(val(substr(ltrim(str(aa,13,2)),11-nr-n;
rn,1)))
    sele 1
    seek int(val(ltrim(str(1))+ltrim(str(n))))
    &or=rtrim(numar_lit)
    if ord=2
        o7=' '
        n7=' '
        o8=' milioane'
        ord=ord+2
    else
        if ord=5
            o4=' '
            n4=' '
            o5=' mii'
            ord=ord+2
        else
            o1=' '
            n1=' '
            o2=' '
            exit
        endif
    endif
endif
    nrn=nrn-1
endd
ni=10
do whil ni#nu
    or='n'+ltrim(str(ni))
    or1='o'+ltrim(str(ni))
    &or=' '
    &or1=' '
    ni=ni-1
endd
* urmeaza executarea corectiilor necesare pen-
* tru o scriere corecta (ex:mie in loc de mii,
```

```
* suta in loc de sute,etc, acolo unde e cazul)
ni = nu
do whil ni#0
    or = 'n' + ltrim(str(ni))
    or1 = 'o' + ltrim(str(ni))
    if &or = 'zero'
            &or = ''
            if (ni = 9.or.ni = 8.or.ni = 6.or.ni = 5)
                    &or1 = ''
            else
                    if ni = 7.and.(n9#' '.or.n8#' '.or.n7#' ')
                            &or1 = 'milioane'
                    else
                            if ni = 4.and.(n6#' '.or.n5#' '.or.n4#' ')
                                    &or1 = 'mii'
                            else
                                    &or1 = ''
                            endif
                    endif
            endif
    endif
    ni = ni-1
endd
if n10 = 'unu'
    n10 = 'un'
    o10 = ' miliard'
endif
if n9 = ' '.and.n8 = ' '.and.n7 = 'unu'
    n7 = 'un'
    o7 = ' milion'
endif
if n6 = ' '.and.n5 = ' '.and.n4 = 'unu'
    n4 = 'o'
    o4 = ' mie'
endif
if n3 = 'unu'
    n3 = 'o'
    o3 = ' suta'
endif
if n9 = 'unu'
    n9 = 'o'
    o9 = ' suta'
endif
if n6 = 'unu'
    n6 = 'o'
    o6 = ' suta'
endif
if n7 = ' '.and.o8 = 'zeci si'
    o8 = 'zeci'
endif
if n4 = ' '.and.o5 = 'zeci si'
    o5 = 'zeci'
endif
if n1 = ' '.and.ltrim(o2) = 'zeci si'
    o2 = 'zeci'
endif
* urmeaza ultima operatie: afisarea partii intregi
* a numarului in litere si a zecimalelor (daca ele
* exista) in cifre/100
if ltrim(substr(str(aa,13,2),12,2))#'00'
    z = ' si ' + ltrim(substr(str(aa,13,2),12,2)) + '/100'
else
    z = ''
endif
clea
? rtrim(n10) + rtrim(o10) + ' ' + rtrim(n9) + rtrim(o9) + ;
' ' + rtrim(n8) + rtrim(o8) + ' ' + rtrim(n7) + rtrim(o7) + ' ' ;
' ' + rtrim(n6) + rtrim(o6) + ' ' + rtrim(n5) + rtrim(o5) + ' ' ;
' ' + rtrim(n4) + rtrim(o4) + ' ' + rtrim(n3) + rtrim(o3) + ' ' ;
' ' + rtrim(n2) + ltrim(rtrim(o2)) + ' ' + rtrim(n1) + rtrim(o:
1) + z
* este refacut vechiul context de lucru
set talk on
* se inchid toate fisierele deschise si se para-
* seste programul
clos all
return
```

O mică glumă

Doriţi să vedeţi conţinutul monitorului d-voastră dansînd pe tact de vals? Dacă veţi rula programul **NUMARULA.PAS**, în scurt timp veţi observa şi efectul. Pe melodia valsului "Dunărea albastră" conţinutul ecranului se va mişca în jos şi în sus. Dar nu vă fie frică, de îndată ce melodia se termină puteţi continua nestingheriţi lucrul - pînă cînd programul îşi va anunţa din nou prezenţa. Pentru a elimina programul din memorie va trebui să iniţializaţi sistemul din nou. În timpul unui test discret, cu colegii, veţi putea observa, fără îndoială, chipuri uimite şi/sau amuzate.

(PC + Technik 1/91, Bastian Schindler)

```
{SM 1024,0,0}
program numarula
uses crt,dos;
var
    Sek, t, z, x, Counter, Sw: Integer;
const Note: array[1..210] of integer =
(294,4,294,4,371,4,441,4,441,4,0,4,882,2,0,2,882,2,0,
6,742,2,0,2,742,2,0,6,294,4,294,4,371,4,441,4,441,4,0,
4,882,2,0,2,882,2,0,6,786,2,0,2,786,2,0,6,278,4,278,4,
330,4,495,4,495,4,0,4,990,2,0,2,990,2,0,6,786,2,0,2,
786,2,0,6,278,4,278,4,330,4,495,4,495,4,0,4,990,2,0,2,
990,2,0,6,742,2,0,2,742,2,0,6,294,4,294,4,371,4,441,4,
589,4,0,4,1178,2,0,2,1178,2,0,6,882,2,0,2,882,2,0,6,
294,4,294,4,371,4,441,4,589,4,0,4,1178,2,0,2,1178,2,0,
6,990,2,0,2,990,2,0,6,330,4,330,4,393,4,495,2,0,2,495,
14,0,2,416,4,441,4,742,16,589,4,371,4,371,8,330,4,495,
8,441,4,294,4,0,2,294,2,294,4,294,6);

procedure MuzicaExtra; interrupt;
begin
dec (Counter);
if Counter <= 0 then begin
    for t:= 1 to 210 do begin
            sound (Note[t]);
            case Note[t] of
                    278,294 : z:= 1; 330 : z:= 2; 371 : z:= 3;
                    395,416 : z:= 4; 441 : z:= 5; 495 : z:= 6;
                    589 : z:= 7; 742 : z:= 8; 786 : z:= 9;
                    882 : z:= 10; 990 : z:= 11; 1178 : z:= 12;
                    0 : z:= 14;
            end;
            z:= z*2-2;
            port[$3d4]:= 13; port[$3d5]:= (z*80) mod 256;
            port[$3d4]:= 12; port[$3d5]:= (z*80) div 256;
            delay (Note[t+1]*4); nosound;
            inc (t,1);
    end;
    counter:= Sw;
    port[$3d4]:= 13; port[$3d5]:= 0;
    port[$3d4]:= 12; port[$3d5]:= 0;
    end;
end;

begin
    val (Paramstr(1),Sek,x);
    if x <> 0 then Sw:= 2184 else Sw:= Sek * 18;
    Counter:= Sw;
    setintvec($1c,@MuzicaExtra);
    keep(0);
end.
```

Mica publicitate

Închiriez XT sau AT cu eventuale servicii de proiectare. Nu pretind plata serviciilor mele timp de un an de la închiriere. Lasuschevici Ion, 1186 Filiaşi, str. Stadionului, bloc K12, ap.4.

Dacă doriţi să utilizaţi calculatorul în rezolvarea problemelor d-voastră, luaţi legătura cu **ELSER SOFT-service,** 1900 Timişoara, str. Circumvalaţiunii nr. 1, tel. 961/25808.

Cumpăr urgent 286 Reference Manual. Popescu Alexandru, 0300 Piteşti, cart. Trivale, bl. P8B, sc.B, ap.6.

Cumpăr Cobra 80k. Ofer 140 DM. Cumpăr Cobra 64k - 100 DM. Cumpăr TV Sport - 5000 lei. Vînd videorecorder 325$. Tel. 921/62152.

Vînd PC AT diferite configuraţii. 90/722460.

Ofer meditaţii şi iniţiere în informatică. Cluj, tel. 40047.

Vînd IBM-PC sau compatibil diferite configuraţii; vînd HC85. Tel. 90/722460.

Informaticieni amatori! În municipiul Tîrgu Mureş fiinţează şi activează clubul Spectrum. Întrunirile noastre au loc în fiecare marţi, între orele 17.30 - 19.00 la sediul nostru din str. Colentina nr.2 apt. 16. Aici aveţi posibilitatea de a face schimburi de programe şi documentaţii pentru mai multe tipuri de calculatoare (Spectrum, Commodore, Apple, IBM PC), de a vă îmbogăţi colecţia proprie cu programe din biblioteca clubului, de a discuta cu alţi pasionaţi ai calculatoarelor personale. Căutaţi-ne cu toată încrederea! ing. Mathe Ştefan, tel. 32594.

Doresc şi ofer **aplicaţii şi programe specifice domeniilor termotehnică, tehnica frigului, transfer de căldură, elementul finit.** Manole Dan Mihai, Facultatea de Mecanică, Catedra de Termotehnică, str. Domneasca 111, 6200 Galaţi.

Realizez **tehnoredactare computerizată** pentru orice fel de cărţi, ziare, reviste. Tel. 263758.

Bronislaw - firmă de informatică efectuează **aplicaţii pe calculator** compatibil IBM PC/XT **în orice domeniu** (programe, editări de texte, etc.). Schimb programe utilitare şi jocuri pentru IBM PC/XT. Bucureşti CP. 12-34.

Izvoditoriul cititorului, sănătate

"Precum de toată probozirea vrédnic să fiu, o, iubitule, foarte bine cunosc (că osteninţa cheltuită nu să jelęşte, fără numai cînd în urmă vreun folos cumva nu aduce)."*

De toată probozirea (mustrarea) vrednici sîntem, cititorule, pentru prea deasa împrumutare a cuvintelor streine. Îţi simţim indignarea ce te cuprinde la fiecare barbarism întîlnit şi-ţi auzim reproşul:

"Nu-s inglez, rumân îmi spune şi să mor rumân eu voi."**

Te înţelegem şi îţi dăm dreptate, cititorule, şi rugămu-te a cerca a nę înţelege şi pe noi. Dificultăţile au fost mari ori de cîte ori "spre céle mai adînci învăţături, prin hirişă limba a noastră a purcéde"* s-a îndrăznit, căci "În privintia terminologiei difficultatile suntu mari"*** şi "de nevoia, unde nu mi a fostu cu potintia de a gassi pene acum, seu a crea unu termenu propriu, nici chiaru dupre consultarea barbatiloru litterati; dera in acestu casu am preferitu sa conservu termenulu nescambatu."***

Te înţelegem şi îţi dăm dreptate şi cînd ne probozeşti că: "Argumentulu co sciinti'a trebue vorbita cu ua limba speciala si neintiellesa multimei este demnu de secolii trecuti de intunerecu si de barbaria si nu aru potea servi astadi decatu ca sa acopere sub velulu misterului nesciintia acellui care cauta sa predea ua sciintia despre care pote nu are de catu idei confuse."***

Ne iartă dară, cititorule, pentru păcatele trecutului, precum şi dacă se va mai întîmpla, şi pentru a mai împuţina din greşalele noastre rugămu-te a ne sări în ajutor cu poveţe şi îndrumări**** pentru ca împreună "toată învăţătura" informaticăi "pre limba noastră în curînd să videm."*

ing. Romulus Maier

---

* D. Cantemir - Istoria ieroglifică, Editura Junimea 1988

** M. Cărtărescu - Levantul, Editura Cartea românească 1990

*** E. Bacaloglo - Elemente de Fisica, Typographia Curtii, Bucuresci 1870

**** talon, if 2/91; mulţumim !

Nimic nu ne sperie! Nimic nu e prea greu! Trei studenţi electronişti în ultimul an de studiu vă oferă orice servicii de programare de la nivelul maşinii pînă la nivelul secretarei. str. Louis Blank nr.2, bl. I1, sc.A, apt. 12, 71.208 Bucureşti 1.

Schimb Windows 2.11 şi DR-DOS 5.0 în limba germană, originale, cu versiunile identice în engleză. Maierean Radu, str. Louis Blank nr.2, bl. I1, sc.A, apt. 12, 71.208 Bucureşti 1.

În numărul viitor:

- Ce este interfaţa IEEE 488?
- Prezentare: 10 compilatoare
- Inteligenţă artificială: reţele neuronale
- Digitizoare

Abonamente la if !

O alternativă la comanda prin ramburs de pe partea cealaltă a talonului şi care simplifică formalităţile de expediţie - abonaţi-vă cu plata în avans! Trimiteţi-ne banii reprezentînd contravaloarea revistelor în contul 40.729.-96.01.04.02 deschis la Banca de Investiţii, filiala Tîrgu Mureş , prin mandat poştal, menţionînd pe cotor (loc pentru corespondenţă...) cîte abonamente şi pe cîte luni doriţi. (Păstraţi chitanţa !) Vă rugăm totuşi să faceţi abonament numai pe lunile rămase pînă la jumătatea anului sau respectiv pînă la sfîrşitul anului. Preţul abonamentului este de 39 lei ori numărul de luni pe care doriţi să vă abonaţi (4 respectiv 10), la care se adaugă cheltuielile de expediţie. Acestea diferă funcţie de numărul de exemplare pe care le abonaţi (de ce n-aţi primi exemplarul dumneavoastră în acelaşi plic cu cel al unui prieten, dacă în felul acesta economisiţi cîţiva lei?), şi funcţie de modalitatea de expediere pe care o alegeţi. Vă putem trimite revistele fie prin coletărie, fie ca scrisori recomandate..
Recomandatele ajung de regulă pînă acasă la abonat, dar costă un pic mai mult; "coletele" trebuiesc ridicate de la poştă şi implică deci un deranj suplimentar, dar sînt mai ieftine. Costul abonamentului, în lei, este sintetizat în tabelul următor:

Nr. exemplare	Recomandat		Prin coletărie	
	4 luni	10 luni	4 luni	10 luni
1	180	450	176	440
2	348	870	332	830
3	516	1290	492	1230
4	680	1700	648	1620
5	848	2120	804	2010

Vom putea lua în considerare abonamente de 4 respectiv 10 luni pînă la data (estimată) a epuizării numărului 3/91: 10 aprilie 1991. După această dată, nu vom fi în măsură să onorăm decît abonamente pe 3 respectiv 9 luni, deci începînd cu numărul 4/91.

Redacţia if

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi lunar |_____| exemplare ale revistei **if,** începînd cu numărul |_____|, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 39 lei, la care se adaugă cheltuielile de expediere. Plata se face prin ramburs. Anularea comenzii se face prin scrisoare recomandată sau cu confirmare de primire şi intră în vigoare din luna următoare.

if 2 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

Preţul unui anunţ este de 10 lei pentru un rînd. Costul unui anunţ se calculează respectînd grila de mai sus - 10 lei per rînd început, indiferent cîte caractere conţine, şi se achită prin mandat poştal pe adresa firmei Micro ATCI, cont 40.729-96.01.04.02 deschis la Banca de Investiţii, filiala Tîrgu Mureş, cel tîrziu la o lună după apariţia anunţului în revistă. Pe mandat (la rubrica "Loc pentru corespondenţă") se va adăuga o explicaţie relativă la suma achitată.
Redacţia nu garantează în nici un fel apariţia anunţurilor achitate anticipat.
Garantez că deţin toate drepturile asupra produselor şi/sau serviciilor ofertate.
Data. ..... Semnătura ..... **if 2 / 91**

Vrînd-nevrînd, problema termenilor "corecţi" în limba română pentru termeni englezeşti care nu au (încă!) echivalent ne urmăreşte. Deoarece nici nu am vrea să fim naşii tuturor noţiunilor noi pentru care nu avem un nume, vă rugăm să ne ajutaţi în găsirea termenilor celor mai adecvaţi, pe care să încercăm să-i promovăm. Uneori nu sîntem consecvenţi nici noi: am propus întîi "chip" pentru"chip", am acceptat pînă la urmă "cip". Dar: să scriem "driver" sau "draivăr",cum se aude? Să acceptăm "magistrală" sau să mergem înainte cu mai simplul "bus"? Să definim OOP ca "programare orientată obiect" sau "programare orientată pe obiecte"? Sîntem sensibili la argumente (necontondente), dar sîntem dispuşi să ne plecăm voinţei majorităţii, dacă se va contura ceva de acest gen.
De data aceasta, am vrea să opinaţi asupra următorilor termeni:

În engleză	româneşte (singular, plural...)
icon	
spreadsheet	
rollbar	
pull-down	
pop-up	
mouse	
object oriented programming	
pushbutton	
state-of-the-art	
cache memory	

Mulţumim !

SOCIETATEA COMERCIALĂ
"TEHNOUTILAJ-G.A." - S.A.
4150 Odorheiu-Secuiesc
jud. Harghita
str. N.Bălcescu nr. 4

pune în vînzare

următoarele echipamente în stare perfectă de funcţionare:

1. Concentrator de date CD-80 compus din:

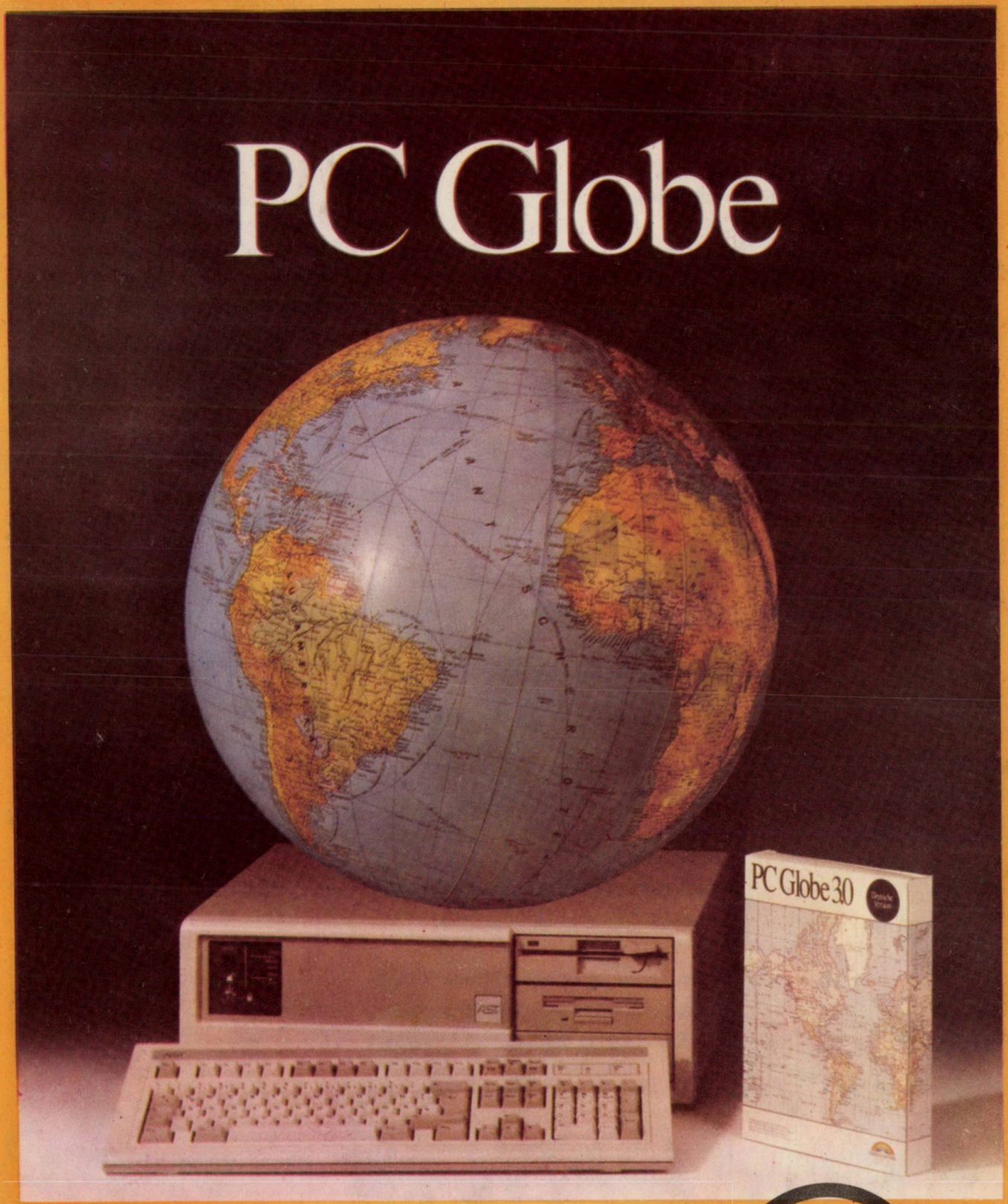
- **M-18**
- **Mx-32 canale- multiplexor de date**
- **unitate banda magnetică tip BW4A4-F**
- **4 buc. VDT 52-RB**
- **2 buc. imprimantă matricială DZM-180**
- **1 buc. consolă Centronics**

2. Minicalculator I-102F compus din

- **unitate centrală**
- **1 buc. unitate disc magnetic 58 Mocteţi**
- **1 buc. unitate bandă magnetică**
- **9 buc. VDT 52-SA**
- **1 buc. imprimantă matricială tip 9335R**
- **1 buc. separator galvanic 16 canale**
- **17 buc. pile de discuri cu 12 sectoare**

Relaţii la telefon:959-11388 int.10423 sau telex: 67411

PC Globe



Markt&Technik

Zeitschriften · Bücher

Software · Schulung

if

CALCULATOARE PERSONALE

Revistă lunară editată de Micro ATCI Tîrgu Mureş

3 / 91

(6)

57 Lei

if if if if

Programe de grafică

Întreţinerea harddisk-urilor

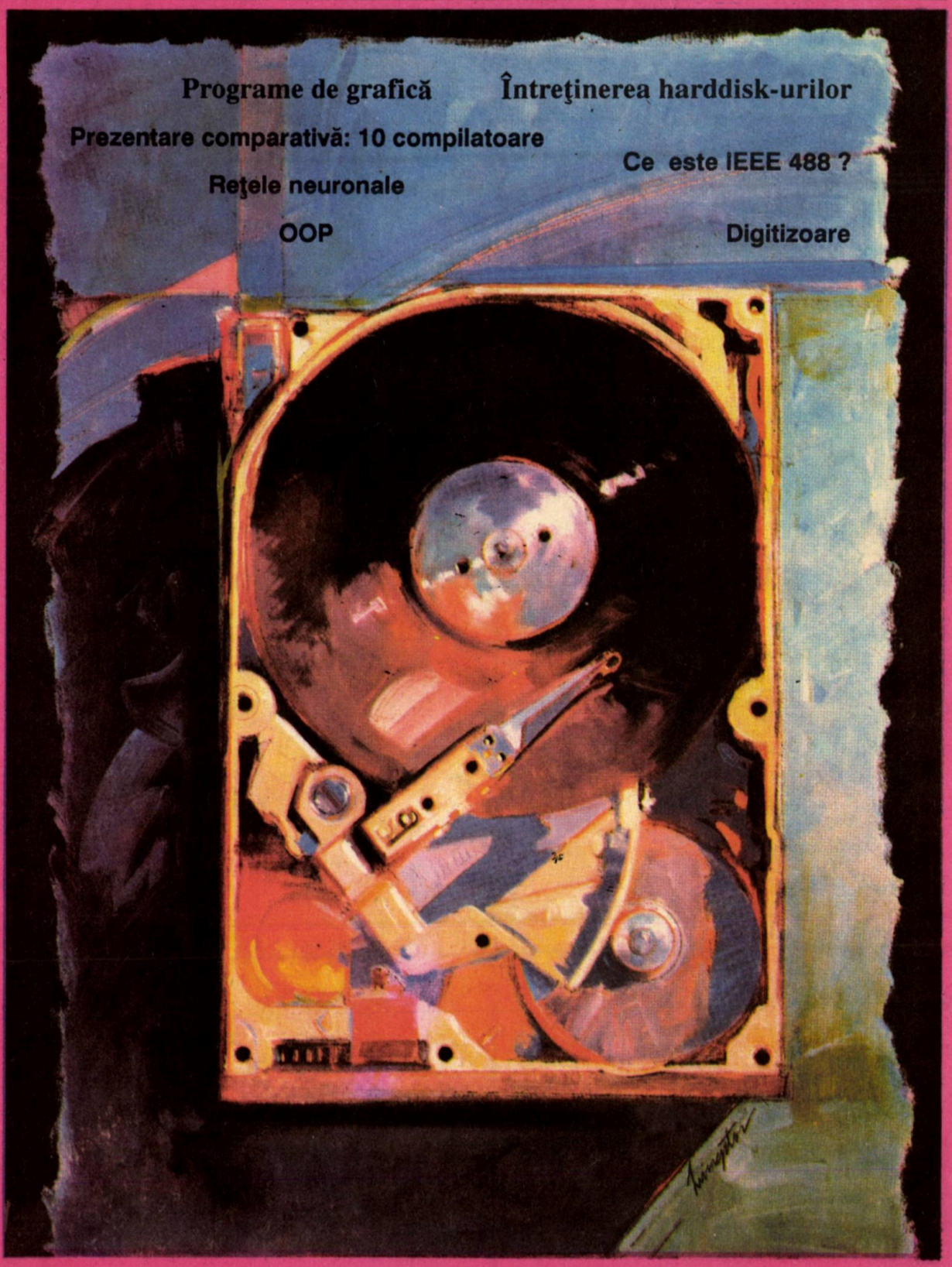
Prezentare comparativă: 10 compilatoare

Ce este IEEE 488 ?

Reţele neuronale

OOP

Digitizoare



AbMod

Ábráhám & Moldovan s.n.c.

va ofera — tehnologie americana, inteligenta europeana, la preturi orientale

Produse din domeniile :
- tehnica de calcul (hardware si software)
- telematica si birotica

Servicii in aceleasi domenii :
- lucrari de cercetare / proiectare
- consulting si asistenta tehnica
- instalare, scolarizare, service
- sisteme "la cheie"

Conditii de livrare : 2 - 4 saptamini

Citeva exemple din lista noastra de preturi :

XT

<< Pret sistem complet 89.000 lei >>
<< cu disk winchester 20 MB 149.000 lei >>

AT - 286 & AT - 386 <<Pret sistem complet [lei]>>

AT	Winchester capacit. formatata	Monitor + interfata MONO 14"	EGA 14"	VGA 14"
286	40 MB	199.000	249.000	299.000
	80 MB	275.000	325.000	375.000
386	80 MB	399.000	-	499.000
	160 MB	549.000	-	649.000

IMPRIMANTE MATRICIALE

Tipul		Pret [lei]
LC 20	80car 9pin	49.000
LC 15	132car 9pin	79.000

Mouse serial compatibil Microsoft 7.700 lei

Copiatoare FC-2 (Canon) 199.000 lei

MS-DOS v4.01 16.000 lei

Cei interesati pot obtine gratuit lista de preturi adresindu-se la:

AbMod Sediul central:
str. Minerilor nr. 42, 3400 Cluj-Napoca Tel. 951 - 56607

AbMod Sucursala Oradea:
str. Decebal nr. 7 bl. D8, ap. 11, 3700 Oradea Tel. 991 - 60278

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

La realizarea acestui număr au colaborat:

ing. Attila Darvas,
ing. Iosif Fettich,
ing. Tibor Kallo
ing. Ingrid Maier,
ing. Romulus Maier,
Colaboratori externi:
mat. Eugen Rotaru
ing. Sava Stan

Tiparul: tipografia Tîrgu Mureş
Tipografi: Halaţiu Mihai,
Cormoş Mircea

Revista apare lunar

Preţ. 57 lei

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/17024, fax 954/35208.

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu directorul. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate la următoarele tarife:
1 cm^2 (alb-negru) 30 lei,
1 cm^2 (alb-negru şi o culoare suplimentară) 40 lei,
1 cm^2 color 60 lei.
Doritorii sînt rugaţi să ia legătura cu redacţia.
Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Sys"if"

"Orice început este greu", dar aşa de greu ca la acest număr, ca la numărul trecut şi ca la celelalte numere care le-au precedat nu ne-a fost la nici un număr. Atunci cînd în urma luptei cu forţele naturii (umane) dezlănţuite mai reuşim să "scoatem" un număr, ne creşte inima de bucurie; am atins culmea. Bucuria este însă de scurtă durată, aidoma lui Sys"**if**", nici n-apucăm să răsuflăm uşuraţi, că ne şi trezim din nou la poalele muntelui. Speranţa că după cîteva urcuşuri vom cunoaşte măcar "traseul" şi că tocindu-ne tălpile vom toci cît de puţin şi din munte, s-a dovedit a fi zadarnică. Dar nu disperăm (probabil că vreun şurubel joacă), şi în loc să ne perfecţionăm cunoştinţele în tehnica vidului (de putere, din stomac, ş.a.m.d.), încercăm să implementăm o revistă de informatică, în spaţiul carpato-danubiano-pontic, pornind din Transilvania şi nu din Transvaal. N-am vrea să fim greşit înţeleşi, noi nu înjurăm pe nimeni, ne-ajunge ce-i în jur.

În acest număr vom descoperi împreună alte tărîmuri ale mirobolantei lumi a calculatoarelor personale. Sperăm că acest prim contact (în **if**) cu informatica de proces, grafica pe calculator şi compilatoarele, să vă ofere răspunsuri la multe întrebări, chiar dacă aceste răspunsuri vor genera şi mai multe întrebări.

Aceste teme, ca şi multe din temele anterior abordate, sînt foarte complexe şi nu pot fi epuizate la o simplă trecere în revistă. Din acest motiv noi vom reveni periodic asupra unor subiecte pentru a le nuanţa şi a observa tendinţele lor de evoluţie.

Începînd din acest număr venim, în sfîrşit, în întîmpinarea dorinţelor formulate de unii cititori cu două cursuri, unul de reţele neuronale şi unul de programare orientată obiect. Acesta este doar un prim pas. Vom încerca să iniţiem, în numerele următoare, şi alte cursuri solicitate. Avînd în vedere intervalul de timp destul de mare dintre două apariţii succesive ale revistei vom încerca să fragmentăm subiectele în aşa fel încît fiecare episod să aibă, pe cît posibil, o existenţă de sine stătătoare.

Criticile, aprecierile şi relatările d-voastră ne vor ajuta să înţelegem unde am greşit (sigur am greşit !), ce am făcut bine (dacă am făcut ceva bine !) şi ne vor permite să ne apropiem mai mult de "ce-i mai mîndru pe la noi". Vă rugăm să nu vă supăraţi pe noi dacă ecoul spuselor d-voastră nu se va face simţit chiar din primul număr care va urma scrisorii. Datorită ritmurilor levantine în care se antamează mai toate pe acest picior de plai, iubit de zei şi bătut de soartă, şi "interstaţiilor amăgirii", de regulă, lucrăm cu două, trei numere în avans. Aşteptăm scrisorile d-voastră !

ing. Romulus Maier

Cuprins

Ştiri

- Cu RISC şi multitasking pag.4
- hobBIT pag.5

Magazin

Înapoi în viitor pag.6

Fabrica de vise are un star nou: calculatorul. Cele mai bune "efecte speciale" ale ultimelor succese cinematografice de la Hollywood i se datorează lui.

Grafică

Programe de grafică pag.8

Artă în 256 de culori pag.11

Formate grafice pag.13

Software

Babilonia limbajelor de programare . . pag.16

- Quick-C cu Quick-Assembler 2.5 . . pag.18
- Quick Basic 4.5 . . . . . pag.18
- Quick-Pascal 1.0 . . . . . pag.18
- Turbo-Pascal 6.0 . . . . . pag.19
- MS-Basic 7.1 - PDS pag.19
- Assembler MASM 5.1 . . . . . pag.19
- Turbo-Assembler 2.0 . . . . . pag.20
- Microsoft-C 6.0 - PDS pag.20
- Turbo C++, Prof. Version 1.0 . . . . pag.20
- Zortech C++, Dev. Edition 2.1 . . . pag.21

Cursuri

Pe drumul spre o tehnică "inteligentă ?" pag.22

Sistemul nervos uman prelucrează mai multe miliarde de informaţii in modul său propriu. Acest mod de lucru al biologicului şi transpunerea sa în sisteme tehnice este astazi parte integrantă a domeniului Inteligenţei Artificiale. Stadiul şi posibilele influenţe ale acestor cercetări vrem să vi le prezentăm în acest curs structurat în patru părti.

Obiectele se încapsulează pag.24

Dacă doriţi să programaţi mai comod şi să lucraţi mai uşor cu proiecte mari, atunci nu puteţi face abstracţie de OOP (programarea orientată obiect).

Hardware

Sisteme de achiziţie de date şi control cu calculatoare personale compatibile IBM PC/XT/AT pag.29

Ce este IEEE 488? pag.39

Digitizoarele - interfeţe spre grafică şi artă pag.42

Reuşita muncii în studiourile CAD, sau în studiourile de grafică, depinde în mod hotărîtor de caracteristicile ergonomice şi de modul de funcţionare al digitizoarelor. Ele sînt interfaţa între mînă şi calculator.

Practică

Să nu uităm CP/M-ul... . . . . pag.44

Sfaturi practice pentru întreţinerea harddisk-urilor pag.48

Reţele

Administratorul unei reţele locale . . . pag.51

Virusologie

Institutul European contra Viruşilor pag.53

Rubrici

- Editorial pag.1
- Poşta redacţiei pag.54
- Mica publicitate pag.54

Cu RISC şi multitasking

Cambridge - Active Book Company a construit un calculator Notepad cu procesor RISC, care umbreşte tot ceea ce s-a realizat pînă acum în acest domeniu. Calculatorul de dimensiunea unui format DIN A4 va fi deservit doar cu un creion electronic.

În laboratoarele de cercetare ale firmei Active Book Company din Cambridge se naşte viitorul. Vă prezentăm un interviu luat de revista PC Magazin directorului acestei firme, Hermann Hauser, în care vor fi prezentate cele mai noi realizări din domeniul Notepads: este vorba despre "Active Book" cu suprafaţa utilizator "Open Book".

PC Magazin: Încă de la mijlocul lui 1990 vorbeaţi despre un prim prototip Active Book. Cît au avansat cercetările ?

Hauser: Dispunem acum de un prim beta-model avînd toate facilităţile funcţionale. Între timp am mai adus cîteva îmbunătăţiri versiunii planificate. Active Book va intra în producţie de serie în al doilea trimestru al anului 1991.

PC Magazin: Pentru ce piaţă aţi conceput calculatorul Active Book?

Hauser: Piaţa spre care ţintim este pe de o parte piaţa verticală a mobililor oameni de afaceri, dar vizăm şi alte pieţe, ca de ex.: sănătate, poliţie, asigurări, personalul vînzător din aşa numitele "Fast moving consumer goods" sau domeniul cercetărilor din domeniul pieţii.

PC Magazin: Vedeţi calculatorul Active Book doar ca pe un Notepad, chiar dacă mai rapid, între celelalte ?

Hauser: În nici un caz. El este mai mult o unealtă productivă pentru manageri din toate domeniile. Lucrări cum ar fi planificarea costurilor, sau asemănătoare, care necesitau adesea mult timp pe un calculator de birou, vor putea fi rezolvate în doar cîteva minute.

PC Magazin: Unitatea centrală a calculatorului Active Book este realizată cu un procesor RISC. Este acelaşi procesor cu care sînt echipate şi calculatoarele Archimedes ?

Hauser: În Notepad-ul nostru am implementat un procesor ARM-2as, numit şi Hercules, un aşa-numit "Full Static Device". Acesta este încă şi mai rapid decît ARM-3, deoarece accesează o memorie RAM statică. Procesorul poate atinge 12.5 MIPS în modul de lucru Burst şi 5 MIPS în medie la o încărcare de durată. Acestea nu sînt însă limitele pe care le poate atinge. ARM-2as poate atinge o viteză de 40 MIPS, dar din motive economice am subdimensionat tactul la care lucrează cipul. Pentru viitor aceasta înseamnă că vom putea, în cazul unor acumulatoare avantajoase, să lăsăm procesorul să lucreze mai rapid, viteza şi puterea fiind în corespondenţă directă.

PC Magazin: De ce nu v-aţi bazat pe tehnicile procesoarelor existente, de ex. Intel ?

Hauser: Decizia de a folosi un procesor RISC ARM n-a fost pusă nici un moment sub semnul întrebării. Ca director şi coparticipant la elaborarea acestui CPU la Acorn sînt convins că doar această tehnică ne va permite să aducem pe piaţă cel mai rapid Notepad. În plus cipul RISC ARM reuneşte pe aceaşi plăcuţă de silicu mai multe funcţiuni: unitatea centrală (CPU) propriu-zisă, RAM-ul static, unitatea de management a memoriei, controller-ele de grafică şi DMA ca şi controller-ele pentru magistralele de 8, 16 şi 32 de biţi. Toate aceste facilităţi sînt realizabile în tehnicile obişnuite de construcţie doar prin utilizarea mai multor module. Avantajul utilizării cipului Hercules este evident: el este singurul procesor din lume care reuneşte într-un singur cip un întreg calculator. Ne-am planificat realizarea unui cip ARM-4 care va fi încă şi mai rapid decît cele mai rapide procesoare ARM de pînă acum.

PC Magazin: Datele culese pe sisteme DOS vor putea fi preluate, prelucrate pe Active Book şi transferate înapoi pe un sistem PC-DOS ?

Hauser: Datele DOS vor putea fi importate deoarece Active Book este compatibil DDE şi DOS. În plus este posibilă utilizarea fie a unui lector de dischete de 3,5" fie a unui harddisk de acelaşi format. Normal că datele vor putea fi preluate în Active Book şi prin intermediul unei interfeţe seriale. Sîntem compatibili cu formatele de date Word, Wordperfect şi Lotus 1-2-3. Cu ajutorul creionului electronic se vor putea, de ex., tăia interactiv celule dintr-o tabelă electronică, introduce valori noi şi altele asemănătoare. Prin utilizarea conceptelor de carte (Book) şi creion (Stift), prin digitalizarea caracterelor se va obţine acelaşi efect ca şi în cazul utilizării unei tastaturi. Aceasta este doar o opţiune, dar noi facem încă un pas mai departe. Olivetti deţine 8 procente în Active Book Company. Dar ce-are a face acest fapt cu compatibilitatea noastră ? Emulatorul nostru de DOS, o soluţie pur soft, este chiar mai compatibil decît, de ex., hardware-ul Olivetti. Acest lucru l-am stabilit cu ajutorul testelor "Validation Suits", pe care Olivetti ni le-a pus la dispoziţie avînd în vederea participarea la afaceri. Aceste sînt teste care verifică compatibilitatea. Pe Active Book poate fi rulat chiar şi simulatorul de zbor al lui Microsoft.

PC Magazin: Ce aplicaţii există pentru Active Book ?

Hauser: Împreună cu acest Notepad livrăm o paletă completă de aplicaţii uzuale, de ex., un program de calcul tabelar, un pachet de grafică, un program de planificare de termene. Aceste soft-uri nu oferă tot atîtea facilităţi ca şi produsele Microsoft, dar prezintă avantajul integrării, deci schimbul dinamic de date între programe atinge o rată mai ridicată. Tocmai a fost abordată funcţia DDE, înafară de aceasta sistemul de operare permite multitasking-ul. El este un sistem Unix, are mărimea de 2 MBytes, se numeşte Helios şi este încărcat în memoria ROM; eu cred că înaintea noastră n-a mai încărcat nimeni atîta software în ROM.

PC Magazin: Ce medii de memorare pot fi utilizate pentru Active Book ?

Hauser: Pe de o parte este opţională utilizarea unui harddisk sau utilizarea unui lector de dischete, pe de altă parte pot fi utilizate diferite memorii de masă ca şi cartele ROM sau PC Memory Cards după standardul PCM-CIA. Cu un harddisk şi cartele de memorie se poate ajunge uşor la mai mult de 100 MBytes memorie de masă.

PC Magazin: Cîte ore poate lucra independent Active Book ?

Hauser: La o utilizare de durată cca. 8 pînă la 10 ore.

PC Magazin: Avînd în vedere utilizarea unei unităţi centrale RISC şi multitudinea facilităţilor oferite, Active Book va fi un Notepad scump ?

Hauser: După calculele noastre preţul se va ridica la cca. 4000 DM. Avînd în vedere caracteristicile de putere putem spune că este avantajos.

(PC Magazin 3/91)

Cu o acţiune de urmărire specială poliţia din München încearcă să rezolve cel mai mare furt de pînă acum din domeniul calculatoarelor. Faptele şi împrejurările în care a fost comis furtul duc la concluzia că faptaşii sînt hoţi profesionişti cu cunoştinţe în domeniu. De la întreprinderea de comerţ CTT din München au fost furate 1110 discuri Winchester şi 25 de adaptoare "Host" în valoare totală de 750.000 DM. Au fost sustrase doar aparate Seagate noi. Spărgătorii nu s-au atins de produsele altor producători şi nici de modelele Seagate reparate a căror ambalaj diferea doar cu puţin de cel al modelelor noi. CTT speră să dea de urma făptaşilor după numerele de înregistrare de pe discuri şi după certificatele de garanţie, de îndată ce aceştia vor ieşi cu ele pe piaţă. Spărgătorii, care au folosit cel puţin o camionetă, au încărcat bunurile în lăzi fără ambalaj ceea ce ar putea duce la deteriorări în timpul transportului. CTT subliniază că nu le-a fost diminuată capacitatea de livrare datorită acestui furt.

Cu versiunea 11 a programului de proiectare asistată de calculator Auto-CAD, Autodesk a adus nu numai o mulţime de facilităţi noi ci s-a hotărît şi asupra unei direcţii strategice: noua versiune va putea fi rulată doar pe PC-uri DOS 386 sau 486 sub SunOS. Alte sisteme de operare, suprafeţe grafice sau platforme (Macintosh, Apollo, VAX) nu vor mai fi luate în considerare. Una din cele mai importante noutăţi ale versiunii 11 este facilitatea restrînsă de lucru în reţea, ce va fi atinsă prin tehnicile Filelocking (blocajul fişierelor) şi Plotspooling (desenare prin spooling). Acum pot fi afişate simultan mai multe vederi în ferestre variabile şi acestea pot fi şi tipărite împreună. Textele de cotare pot fi afişte în culoarea şi la înclinarea dorită în cazul unui desen.

În sfîrşit: o revistă pentru posesorii de HC-uri!

hobBIT

se numeşte noua apariţie; revista este editată de Clubul Român de Calculatoare (P.O.Box 37-131 Bucureşti) şi conţine, pe parcursul a 32 de pagini, informaţii despre CRC, Beta Basic, Simon's Basic, o pagină pentru începători, jocuri comentate, tips & tricks, o pagină pentru constructorii amatori şi altele. Revista costă 35 lei plus 4,80 lei taxă de expediţie.

Seminar

În perioada 6-8 februarie şi respectiv 12-14 februarie, la Bucureşti şi Sibiu a avut loc seminarul "Micro-Informatica, Sisteme şi Management" organizat de firma LOGIC sub egida CNI.
Spicuim din tematica seminarului:

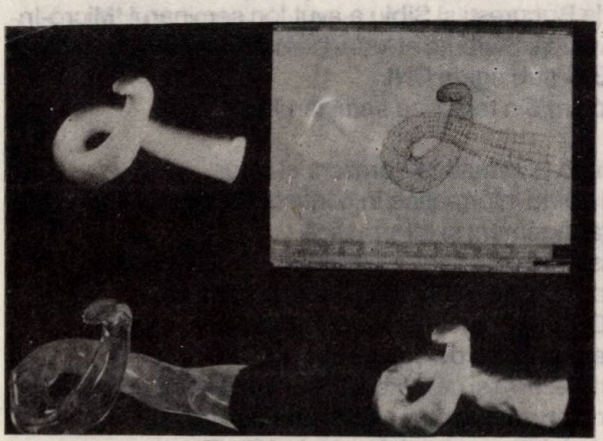
- Micro-Informatica - prezent şi viitor
- Sisteme informatice în contextul economiei de piaţă
- Management şi Informatica
- Sisteme de operare pentru microcalculatoare
- Baze de date şi limbaje de programare
- BORLAND: prezentarea produselor program
- Reţele locale de calculatoare (LAN)
- Arhitectura şi caracteristicile reţelelor de tip NOVELL
- Configuraţii pentru reţelele de tip NOVELL
- Administrarea reţelelor de tip NOVELL

Înapoi în viitor

Fabrica de vise are un star nou: calculatorul. Cele mai bune "efecte speciale" ale ultimelor succese cinematografice de la Hollywood i se datorează lui.

Apa trăieşte: oamenii care îşi aşteaptă ultima salvare într-o staţie subacvatică nu-şi pot crede ochilor. Din apa liniştită a ecluzei se ridică un sorb care se îndreaptă în mod evident spre atractiva comandantă a submarinului. În momentul în care creatura marină a ajuns la înălţimea chipului femeii, îi reproduce fizionomia - 100 de litri de apă de Pacific îi surîd. Cel mai tîrziu în acest moment, cinefilul se simte cuprins de afecţiune pentru o masă de apă. Aceasta este cea mai bună dovadă pentru funcţionarea trucului.

Întîlnirea cu creatura acvatică face parte din filmul pentru copii "Abyss", care în 1989 a produs mare vîlvă. Motivul: acţiunea foarte palpitantă, înscenată de regizorul James Cameron (Aliens), a fost condimentată cu efecte inexistente pînă acum. Acestea provin în bună parte din calculator, căci cu ajutorul efectelor speciale uzuale asemenea trucuri, cum ar fi coloana de apă inteligentă, nu ar fi posibile. Calculatorul îşi face intrarea în fabrica de vise.



Monstrul marin a fost creat pe calculator

"Grafica pe calculator potenţează creativitatea. În momentul în care un regizor trebuie să se întrebe mereu dacă ideile sale vor putea fi traduse în viaţă, ajunge să se cenzureze singur. Dar dacă orice există în fantezia sa poate fi realizat pe ecran, atunci el se poate concentra asupra acţiunii filmului şi poate crea o lume de vis." ne spune Doug Kay directorul departamentului de grafică pe calculator al firmei ILM ("Industrial Light and Magic"), firma cu cei mai buni specialişti în materie de trucuri din lume.

În spatele firmei ILM stă o legendă vie a Hollywood-ului: producătorul George Lucas, care a realizat, împreună cu regizorul Steven Spielberg, filme de mare succes, cum ar fi: "Războiul stelelor", "ET" şi "Indiana Jones". Secţia de trucuri, înfiinţată în 1981, trebuie să răspundă unor mari pretenţii: ILM trebuia să producă, pentru Lucas şi Spielberg, iluzii perfecte. Tot ceea ce este prea costisitor de realizat, sau prea periculos pentru cascadori, este făcut de ILM. "Specialitatea noastră este imposibilul", afirmă Doug Kay.

Chiar şi monstrul marin din Abyss părea de nefilmat. Problema: apa este transparentă, deformează imaginea din fundal şi aruncă pete de lumină pe pereţi. Deoarece se dorea ca scena să pară reală, nu se putea filma pe baza trucurilor uzuale, cu modele sau cu păpuşi. "Nici specialiştii noştri în butaforie n-au descoperit încă apa care să fie în acelaşi timp şi solidă şi fluidă", ne spune Doug Kay. Din acest motiv specialiştii în trucuri au optat pentru grafica pe calculator, deoarece calculatorul este în stare să calculeze la fel de bine imaginea unei coloane de apă ca şi pe cea a unei bile de crom sau a unei mese de lemn.

Problema era cum să se introducă monstrul, în mod perfect, în mediul real al filmului. "Monstrul trebuia să arate natural, deci pereţii submarinului trebuiau să se reflecte în apă." descrie Doug Kay una din problemele principale. Dar cum se pot introduce culisele deja filmate în calculator ?

Soluţia este dată de utilizarea unui scanner cu laser, care înregistrează cadrele filmului unul după altul, şi le înmagazinează în memoria calculatorului. Dacă imaginile de bază se află în calculator, specialiştii pot modifica imaginile după dorinţă.

În Abyss, calculatorul a calculat, pe baza formulelor din fizică, în mod exact şi corect, reflexiile luminii, aşa cum ar fi fost văzute dacă monstrul ar fi existat în realitate. Imaginea completată, în calculator, cu coloana de apă va fi transferată cu ajutorul unei imprimante laser color pe o nouă peliculă de film, şi astfel efectul special este gata.

Colectivul a lucrat intens patru luni, pînă cînd detaliile au corespuns în totalitate şi pînă cînd regizorul James Cameron a fost convins. "N-a crezut că aşa ceva s-ar putea realiza cu calculatorul. A trebuit să-l convingem că monstrul va părea real şi nu va arăta ca o grafică ieftină pe calculator, şi am reuşit." spune Doug Kay.

Cea mai uluitoare aplicaţie a manipulării imaginii: pe calculator reuşesc metamorfoze la care nici pictorul M. C. Escher n-ar fi putut visa - dintr-o clipă într-alta un om se poate transforma într-un leu, sau o capră se poate transforma într-o

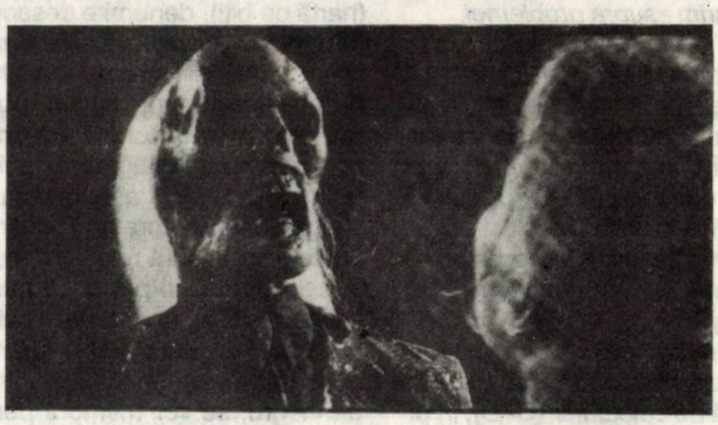
pasăre. Doug Kay declară: "Cu calculatorul putem transforma orice figură în oricare altă figură. Pentru aceasta avem nevoie doar de imaginea iniţială şi de cea finală - restul îl rezolvă programele noastre."

Cu acest truc a fost realizat şi teribilul final din filmul "Indiana Jones şi ultima cruciadă". Ticălosul Donovan vrea să bea din izvorul sfînt, cu un pahar fermecat, descoperit de Indiana Jones, pentru a deveni nemuritor, dar alege greşit paharul şi se transformă în praf în cîteva secunde. Doug Kay ne povesteşte: "În mod normal pentru această scenă actorul şi trei machiori ar fi fost ocupaţi mai multe zile. Pentru trei secunde de film ar fi trebuit făcute mai mult de 150 de modificări de machiaj. Într-o astfel de situaţie calculatorul este mai rentabil şi menajează şi nervii colectivului."

Tehnicienii ILM modifică, cu ajutorul calculatorului, nu numai imagini, ei îl folosesc şi pentru mixajul digital de imagini. În filmul "Înapoi în viitor", actorul principal Michael J. Fox stă de patru ori cu el însuşi la masă, de fiecare dată în alt rol. Pentru aceasta fiecare poziţie a fost filmată separat, iar calculatorul a copiat într-o scenă imagine cu imagine, astfel încît diferitele variante Fox să se suprapună parţial. Calculatorul este în mod automat atent ca perspectivele să corespundă, astfel ca o persoană din prim plan să pară mai mare decît una din fundal.

Pentru mixarea unor imagini trucate şi a unor imagini reale, se foloseşte de secole tehnica "blue box". Toate scenele se înregistrează pe fundalul unui cer albastru. Prin metoda "blue box" fundalul poate fi mascat. Tot ceea ce este albastru va dispărea. Prin suprapunere se pot mixa două imagini diferite.

În studiourile ILM, izolate ermetic, se află cel mai mare ecran albastru din lume. El este mai înalt de



Nu machiorii ci calculatorul este cel care-l transformă pe Donovan în praf în filmul Indiana Jones şi ultima cruciadă

15 metri şi lat de 20 de metri, şi este suficient pentru a face o maşină să zboare, în filmul "Înapoi în viitor", sau pentru a face să zboare trei copii pe biciclete, în "ET".

Indiferent de cît de perfecte apar trucurile specialiştilor, o greşeală, cît de mică, tot se strecoară în fiecare film. Foarte rar însă ea este atît de crasă ca în partea a doua a filmului "Războiul stelelor". Chiar la început apare pe ecran, cu litere imense, titlul filmului: "Imperiul contraatacă". Este o greşeală clasică de scriere de care calculatorul nu este vinovat. Nobody is perfect, nici chiar George Lucas.

(Computer Live 9/90, Gregor Neumann)

Programe de grafică

Denumirea de programe de grafică ascunde o întreagă lume. Şi pentru că această lume este atît de diversă, alegerea unui program anume este dificilă. Cine spune doar că doreşte să cumpere un program cu care să deseneze grafice, poate călca în strâchini. Să dăm deci strâchinile la o parte şi să ne lămurim asupra problemei.

Programele de grafică sînt atît de bogate în faţete, încît unele programe nici nu pot fi încadrate corect într-o categorie anume. Dar care sînt categoriile ? Programele de grafică s-ar putea clasifica în: programe de pictură, programe de desenare, programe pentru ilustrare, grafică de afaceri, pachete de prezentare evoluate, animaţie în două dimensiuni (2D), programare asistată de calculator (CAD), în diferitele sale variante (de la programe 2D orientate pe muchii pînă la modelarea solidelor), programe de prelucrare a imaginii, programe de "umbrire" (Rendering Software) şi de "iluminare" (Raytracing Software), programe de trasare automată şi tot ceea ce poate constitui o combinaţie a celor de mai sus, sau care se află la limita a două sau mai multe categorii. Cel care a reuşit să depene acest ghem şi a înţeles fiecare termen va găsi programul de grafică dorit mult mai uşor.

Marea diferenţă

Formatele grafice pot fi clasificate în două grupe principale:

- prima cuprinde formatele grafice vectoriale, respectiv pe cele orientate pe obiecte.
- a doua cuprinde formatele grafice orientate pe pixeli, sau bitmap.

În cazul programelor grafice vectoriale, (numite şi "programe de desen") liniile, pătratele sau cercurile sînt tratate ca şi obiecte, iar toate informaţiile despre imagine sînt administrate de programe sub forma unor descrieri matematice ale obiectelor, în acest mod fiind şi salvate pe dischete sau harddiskuri.

Programele grafice bitmap (hartă de biţi), denumite deseori şi "programe de pictură", înmagazinează şi administrează imaginile descriind amănunţit fiecare punct al imaginii. În cazul lor, un grafic este înţeles ca o imagine construită pe rînduri, în mod asemănător cu imaginea de pe ecranul TV, care şi ea este construită rînd cu rînd, iar în cadrul rîndului punct cu punct. În acest caz se memorează culoarea fiecărui punct, într-o anumită succesiune, iar în cazul imaginilor alb-negru, se vor memora pentru fiecare punct informaţiile "punct aprins" (alb) sau "punct stins" (negru).

Ambele tipuri de programe au atuuri puternice şi deci şi domenii speciale de utilizare. Programele de grafică vectorială au fost dezvoltate iniţial pentru domeniul construcţiilor (proiectare asistată de calculator PAC - CAD în original: Computer Aided Design), dar datorită creşterii continue a performanţelor lor au ajuns să aibă întîietate în tot mai multe domenii. În domeniului design-ului profesional aceste programe sînt utilizate aproape în exclusivitate.

Utilizarea lor este avantajoasă deoarece operînd cu obiecte, desenele pot fi foarte uşor modificate şi corectate (mult mai uşor decît la programele orientate pe pixeli). Spre exemplu, orice obiect de pe suprafaţa de desen este liber, poate fi mişcat liber, este scalabil (poate fi mărit sau micşorat), poate fi rotit, i se poate defini în mod liber culoarea (totul depinde doar de posibilităţile programului lansat). În plus, oricare ar fi imprimanta folosită pentru tipărire, desenul va fi tipărit de fiecare dată cu rezoluţia şi fineţea maximă oferită de imprimantă.

Punct cu punct: cui foloseşte ?

La programele grafice bitmap, situaţia se prezintă altfel. Ele nu se pretează la construcţia de planuri sau imagini, deoarece fiecare linie sau cerc care sînt desenate rămîn (aproape) definitive. Majoritatea programelor posedă, de cîtva timp, şi funcţia UNDO, cu ajutorul căreia desenatorul poate anula ultima operaţie efectuată. Dar dacă, din greşeală, aţi desenat un pătrat negru peste un desen fin şi complicat, atunci nu se mai poate repara nimic.

La programele orientate pe obiecte desenul din linii se mai poate vedea şi după ce pătratul negru a fost îndepărtat. Nici la tipărirea pe imprimantă imaginile bitmap nu pot fi tipărite atît de flexibil ca şi cele vectorizate. Dacă, de ex. desenăm cu un program bitmap un cerc cu un diametru de 100 de puncte, atunci acest cerc va fi tipărit ca atare. Dacă mărim graficul, ceea ce cu un program DTP se poate face foarte uşor, atunci spaţiul ocupat de grafic va creşte, bineînţeles, dar rezoluţia va rămîne aceeaşi şi calitatea imaginii va scădea, iar prin mărire succesivă treptele graficului (care apar datorită punctelor dreptunghiulare, singulare ale imaginii) vor fi din ce în ce mai vizibile.

Dar unde sînt atunci avantajele programelor bitmap ? În primul rînd în domeniul artei pe calculator. Programele bitmap pot reprezenta şi prelucra un număr tot mai mare de culori. Standardul actual, în domeniul profesional, atinge cifra maximă de 16.7 milioane de culori. Aceasta este suficient chiar şi pentru profesioniştii cei mai pretenţioşi.

Un alt domeniu, foarte important, de utilizare a programelor grafice orientate pe pixeli este prelucrarea şi retuşarea imaginilor, care sînt preluate pe calculator prin intermediul unui scanner sau camere video. Acest domeniu este foarte important pentru utilizatorii care doresc să preia imagini, deja existente, în documentele DTP.

Categoriile programelor de grafică

Programe de pictură

Programele de pictură sînt programe cu ajutorul cărora pot fi desenate imagini orientate pe puncte. Dintre acestea amintim: Deluxe Paint II PC, PC Paintbrush, GEM Paint, PC Atelier, Colorix VGA Paint. Tot din această categorie fac parte şi programele de retuşare a imaginilor scanate. Programele de pictură servesc în primul rînd la pictarea de imagini şi în al doilea rînd la prelucrarea lor.

Programe de prelucrare a imaginilor

Aceste programe sînt o formă mai dezvoltată a programelor de pictură. În cazul lor, nu funcţiile de desenare sînt importante, ci funcţiile de prelucrare a imaginilor deja existente. Se pune un accent deosebit pe prelucrarea fotografiilor de rezoluţie înaltă scanate. Cele mai importante funcţii sînt cele de reglare a contrastului, de montaj a imaginilor şi cele care permit obţinerea de efecte fotografice. Programele de prelucrare a imaginilor au nevoie, de obicei, de multă memorie şi în plus necesită plăci grafice cu trepte de gri de rezoluţie maximă. Cel mai cunoscut produs, din această categorie, sub MS-DOS, în momentul de faţă este "Imagestudio Windows". Deoarece aceste programe necesită anumite condiţii speciale, ele se vînd în majoritatea cazurilor împreună cu cartelele grafice corespunzătoare, ca soluţii complete.

Grafică de afaceri şi de prezentare

Aceste programe produc în primul rînd reprezentări grafice ale unor mulţimi de cifre, pe scurt grafice (charts). Aceste grafice pot fi fie diagrame tort, fie histograme, grafice plasă, etc. Pachetele soft Microsoft Chart, Harvard Graphics, Mirage sau Perspective, sînt în primul rînd programe de desenare de grafice. Mai nou toate programele de calcul tabelar au integrate şi funcţii de grafică de afaceri. Unele programe s-au specializat pe grafice tridimensionale, altele s-au orientat spre grafica de prezentare. În unele programe de prezentare au fost preluate funcţii de aranjare în pagină, ca şi cele de la programele DTP, şi au fost preluate chiar şi funcţii specifice programelor de pictură şi de desen. Unele conţin rutine de prezentare (show) şi rutine de animaţie. Diferenţele dintre caracteristicile de performanţă în cadrul acestei categorii sînt considerabile.

Programe de animaţie

Cu ajutorul acestor programe se poate produce grafică animată. Cele mai multe dintre aceste programe sînt comandate de limbaje Script, de exemplu Grasp şi VCN Concorde. Grasp atinge puţin şi categoria grafică de prezentare. Cel mai nou şi mai eficient program din acest domeniu este Autodesk Animator, care conţine diferite tehnici de animaţie, cum ar fi: "page-flipping" - alternarea de imagini întregi, ca la desene animate, "cel animation" - mişcarea doar a unor părţi ale imaginii, în timp ce partea fixă a imaginii rămîne neschimbată, "tweening" - fazele intermediare între două stări ale unui obiect sînt calculate în mod automat de către program şi cuplarea cu imagini umbrite.

Programe de desenare

Programele de desenare sînt programe simple orientate pe obiecte, fără funcţii speciale suplimentare. Reprezentanţii standard ai acestei categorii sînt: GEM Draw Plus, sau Micrografx Draw Plus. Ele nu oferă nimic extraordinar, cum fac programele CAD, sau cele de ilustrare, dar tocmai din acest motiv sînt şi mult mai ieftine. De regulă funcţiunile acestor programe sînt suficiente pentru a produce ilustraţii simple, diagrame de flux, grafice tehnice puţin complicate, şi altele asemenea.

Programe de ilustrare

De regulă programele de ilustrare sînt versiuni dezvoltate ale programelor de desen, unele sînt însă concepute în mod special pentru prelucrarea unor grafice obţinute cu programe de desen. Programele de ilustrare sînt utilizate mai ales pe piaţa DTP (Desktop Publishing) şi ele pot fi folosite, printre altele, şi pentru producerea graficelor pentru cataloage. Programele de ilustrare stăpînesc separarea culorilor în cele patru culori de bază ale imprimantelor color, funcţiile lor de desenare sînt specializate pe ilustrarea cu contururi rotunjite (spre deosebire de programele de desenare care de regulă tratează doar obiecte colţuroase). Concurenţa între produsele acestei categorii este foarte mare, astfel încît la fiecare trei luni apare pe piaţă cîte o nouă versiune, mai performantă, a unuia sau a altuia dintre produse. Cele mai noi în momentul de faţă sînt Designer 3.01 şi Corel Draw 2.0, dar tot din această categorie fac parte şi Diagraph Windows şi Artline.

Programe CAD

CAD a fost la început prescurtarea pentru "Computer Aided Drafting" (desenare asistată de calculator) şi a devenit mai tîrziu prescurtarea pentru "Computer Aided Design" (proiectare asistată de calculator). Aceste programe au fost gîndite în primul rînd pentru scopuri de construcţii. Deoarece totul trebuie reprezentat cu o precizie matematică, funcţiunile programelor CAD sînt specializate pe

desene tehnice. Deci ele deţin funcţii de măsurare, permit introducerea coordonatelor exacte, tipărirea la scară, etc. De obicei ieşirea se face pe plotter, obţinîndu-se în final desenele tehnice de execuţie. Majoritatea programelor CAD dispun şi de drivere pentru imprimante laser sau matriciale.

Există programe CAD specializate pentru proiectarea cablajelor imprimate, pentru arhitectură, arhitectură internă, construcţii de maşini, design auto, şi chiar şi unele specializate pe decoraţiuni interioare . Cel mai cunoscut program CAD este Autocad care a acaparat mai mult de 40% din piaţă.

Sistemele CAD se mai pot clasifica în sisteme 2D şi în sisteme 3D. Sistemele CAD 2D pot fi la rîndul lor orientate pe muchii sau pe elemente. Există diferite realizări de sisteme 3D, de la cele simple, tip model din sîrmă, pînă la sistemele orientate pe suprafeţe sau pe volume, numite şi de modelare a solidelor ("Solid Modeling"). Sistemele orientate pe volume sînt răspîndite mai ales în domeniul construcţiilor mecanice, deoarece cu ajutorul lor se pot calcula modificările de volume, sau greutate, ale unor piese după o prelucrare, de ex. ştanţarea unei găuri. Aici CAD se apropie foarte mult de domeniul fizicii ştiinţifice.

Programe de desenare orientate CAD şi programe CAD pentru începători

Aceste programe sînt un amestec de programe de desen şi programe CAD. Din această categorie fac parte: Autoskétch, Gfa-Draft, In-a-Vision, etc. În SUA această categorie este denumită "CADD", primul D venind de la Drafting, deci desenare, iar al doilea D vine de la Design, dar ar trebui să fie doar o jumătate de D, deoarece partea de proiectare nu este tot atît de performantă ca şi cea a sistemelor CAD profesionale.

Aceste produse sînt destul de bune pentru lucrările simple de construcţii. Unele produse nu sînt altceva decît versiuni simplificate ale programelor CAD profesionale, care permit utilizatorului să facă primii paşi în această lume la un preţ acceptabil. Aşa s-au născut de exemplu Easycad şi Fastcad. Altele, ca In-a-Vision de exemplu, au fost obţinute prin dezvoltarea unor programe de desenare (în cazul nostru din Windows Draw). Uneori aceste programe pot fi folosite tot atît de simplu ca şi programele de desen, alteori sînt la fel de greoaie ca şi programele CAD.

Programe de umbrire (Rendering) şi de iluminare (Raytracing)

Cele mai multe programe CAD de modelare a solidelor posedă deja şi funcţii de umbrire. În acelaşi scop pot fi folosite şi produse specializate, cum ar fi Autoshade. Ele se folosesc, de regulă, pentru a da impresia că obiectele construite sînt reale.

Majoritatea programelor de umbrire sînt oferite ca programe auxiliare unor programe CAD. Deoarece ele sînt folosite de obicei de arhitecţi care vor doar să privească rezultatul, rutinele de umbrire trebuie să fie rapide. Din acest motiv se folosesc agoritmii de umbrire "Phong" şi "Gouraud", care produc imagini semireale.

Programele de iluminare au nevoie sensibilă de mai mult timp. Ele calculează pentru fiecare punct de pe o suprafaţă modul de reflectare al luminii şi nuanţele de culoare ale umbrei acestui punct. Această metodă oferă imagini mult mai reale decît celelalte metode, dar este gîndită pentru a fi utilizată mai ales "peste noapte".

Programe de trasare automată (Autotracer)

Aceste programe sînt specializate în vectorizarea desenelor orientate pe puncte. Ele sînt capabile astfel să tipărească şi imaginile scanate cu o rezoluţie foarte mare, de exemplu pentru o instalaţie tipografică. Majoritatea programelor de ilustrare conţin şi funcţii autotrace. Un produs nou, Adobe Streamline, s-a specializat pe trasare şi poate transforma în grafice orientate pe obiecte, chiar şi imagini complexe. Înainte de a folosi rezultatul trasării, aceste imaginii mai sînt prelucrate, de regulă, cu un program de ilustrare.

Denumirea vine de la cuvintele "auto" (automat) şi "trace" (urmărire, trasare). La copierea unei imagini cu mîna se urmăreşte conturul imaginii aflate sub folia pe care se copiază. În mod asemănător lucrează şi aceste programe.

Trasarea automată, autotracer, nu trebuie confundată cu găsirea automată a traseului, autoroute. Programele profesionale de proiectare a cablajelor imprimate deţin de obicei o funcţie autoroute, care caută în mod automat cea mai bună legătură (cel mai bun traseu), între două puncte de pe placă.

Şi restul ...

În această enumerare nu au fost amintite multe utilitare şi programe experimentale. Bineînţeles că o parte din ele nici nu pot fi categorisite corect. Unele produse sînt oferite de producător cu eticheta "performant CAD", cu toate că performanţele amintesc mai degrabă de un program de desen simplu.

Multe produse rămîn în umbră, cu toate că sînt bune, deoarece mulţi cumpărători încearcă să le încadreze între nişte limite clare. Trasarea limitelor se dovedeşte însă a fi o greşeală - nu vă alegeţi softul doar după categorie ci şi după caracteristicile de performanţă. Astfel, de ex., un program de desenare cu funcţii CAD simple poate face uneori servicii mai bune decît un sistem CAD, greu de învăţat. Un alt exemplu: combinaţia dintre un program de grafică economică şi unul de pictură, suficientă pentru multe scopuri costă de multe ori doar pe jumătate din cît costă aşa-numitele programe de prezentare. Este bine deci să reflectaţi de două ori înainte de a vă cufunda în "haosul graficii".

(R.M.)

Artă în 256 de culori

Programele de pictură nu au nici o utilitate. Ele nu sînt de folos nici planificării bugetului, nu produc nici scrisori de afaceri şi nu ne amintesc nici termene importante. În multe alte privinţe, ele depăşesc însă aplicaţiile "serioase" de pe PC-uri, ele dau mai multă satisfacţie. Pe primul loc în această categorie se află "Deluxe Paint II", al lui Electronics Arts, care în versiunea extinsă este denumit "Enhanced". Cei mai importanţi concurenţi sînt "VGA Paint" al firmei Rix Softworks şi "PC Paintbrush IV Plus" al firmei ZSoft. Cele trei programe au un singur lucru în comun: ele folosesc ecranul PC-ului ca pînză şi mouse-ul ca pensulă.

Deluxe Paint poate lucra cu o paletă de 256 de culori, dar rezoluţia pe care o poate atinge este de doar 320X200 de pixeli. Concurenţii săi pot atinge o rezoluţie de pînă la 640X400 de pixeli. VGA Paint mai oferă şi posibilitatea de a se lucra cu rezoluţia, nedocumentată, de 360X480 de pixeli. Acest lucru este util atunci cînd placa VGA nu are o rezoluţie de 640X400 de puncte, în acest caz la nevoie se poate lucra şi cu rezoluţia mai slabă de 320X200 de puncte.

La pornirea programelor, cîteva diferenţe sînt frapante: în timp ce Deluxe Paint şi PC Paintbrush lucrează cu tehnica ferestrelor şi cu meniuri Pull-Down, VGA Paint foloseşte meniurile Pop-Up. Aceasta înseamnă că pe ecran este vizibilă doar grafica, meniurile apărînd doar la cerere, în timp ce la Deluxe Paint şi la PC Paintbrush rîndurile de meniu şi "icon"-urile sînt mereu prezente. Acest lucru are atît avantaje cît şi dezavantaje: pe de o parte structura meniurilor şi deci modul de utilizare este mereu la dispoziţie, pe de altă parte ecranul nu poate fi folosit în întregime. Acest neajuns se observă mai ales la Deluxe Paint, suprafaţa utilizator poate fi văzută în întregime doar pentru afişarea unei imagini, pictarea obiectelor care ar putea umple ecranul este imposibilă, datorită suprafeţei mici puse la dispoziţie pentru desen. Aceste obiecte sînt practic tăiate la marginea ferestrei.

Mai importante decît aceste lipsuri sînt însă funcţiile de desenare pe care le oferă fiecare program în parte. La acest capitol Deluxe Paint se află în frunte. Datorită multitudinii posibilităţilor de manipulare, PC Paintbrush şi VGA Paint rămîn privitori de pe margine. De ex. imaginile pot fi îndoite cilindric, pot fi întoarse şi răsucite, paleta de culori poate fi animată (poate fi parcursă ciclic) şi fiecare desen (porţiune de desen) poate fi multiplicat simetric cu funcţia de caleidoscop. Cel mai important lucru este însă modalitatea de combinare a tuturor funcţiilor. Prin aceasta chiar şi cele mai complexe modificări pot fi făcute în modul zoom (efect de lupă), şi fiecare funcţie poate fi cuplată cu un Anti-Aliasing-Effect cu ajutorul căruia pot fi şterse treptele colţuroase de pixeli. Avantajul este clar: mai mulţi paşi individuali pot fi reduşi la unul singur.

PC Paintbrush aduce, în schimb, alt avantaj şi anume posibilitatea de a prelua imagini prin intermediul unui scanner. Această posibilitate lipseşte atît la Deluxe Paint cît şi la VGA Paint. Ca şi număr de funcţii PC Paintbrush nu poate ţine pasul cu Deluxe Paint. El este mai bun însă pentru desenarea unor detalii mici dar fine. El are posibilitatea de mişcare limitată a cursorului orizontal şi respectiv vertical, astfel încît se pot desena linii drepte orizontale sau verticale chiar şi cu mîna liberă. Şi funcţia de introducere text este interesantă. Textele pot fi prevăzute în mod automat cu umbre şi culori, ceea ce în cazul celorlalte două programe nu se poate realiza decît cu mari eforturi.

VGA Paint se deosebeşte de cei doi concurenţi ai săi în două puncte: este singurul care pune la dispoziţie întregul ecran pentru desen şi chiar şi pe cele mai mici plăci grafice VGA pune la dispoziţie un mod de lucru care permite folosirea unei palete de 256 de culori la o rezoluţie mult mai bună decît învechita 320X200.

În concluzie se poate preciza că utilizatorii unor plăci grafice VGA mici ar putea folosi VGA Paint, în timp ce PC Paintbrush ar trebui folosit atunci cînd trebuie scanate şi prelucrate imagini deja existente. În sfîrşit, Deluxe Paint este cel care le depăşeşte pe celelalte două, nu pune limite creativităţii utilizatorului, multitudinea posibilităţilor de a obţine efecte speciale deschide perspective cu totul noi, iar preţul este inferior celui al concurenţei.

(Computer Persönlich 25/90, Matthias Fichtner)

Tabel comparativ

Program	Deluxe Paint II Enhanced	Colorix VGA Paint	PC Paintbrush Plus
Date tehnice			
EMS	recomandabil	nu recunoaşte	va recunoaşte
Placa grafică	HGC, CGA, EGA, (E-)VGA	VGA	HGC, CGA, EGA, (E-)VGA
Rezoluţia maximă	1024X768 puncte (16 culori)	1024X768 puncte (16 culori)	1600X1200 puncte (2 culori)
Nr. maxim de culori	256 (la 640X480 puncte)	256 (la 800X600 puncte)	256 (la 1024X768 puncte)
Funcţii de desenare			
Mînă liberă	da, 3 variante	da, 2 variante	da, 3 variante
Punct	da	nu	nu
Linie	da, 2 variante	da	da, 3 variante
Curbe	da, 2 variante	da	da, 3 variante
Cerc/Elipsă	da, 6 variante	da	da, 8 variante
Dreptunghi	da, 4 variante	da	da, 4 variante
Poligon	da, 3 variante	nu	da, 2 variante
Umplere	da, 10 variante	da	da, 2 variante
Pulverizare	da, şi cu conţinutul pensulei	da	da
Funcţii de secţionare şi pensulă			
Secţionare dreptunghiulară	da	da	da
Secţionare cu mîna liberă	da	nu	da
Mărire/micşorare	da	da	da
Basculare	da	da	da
Rotire	da	da	da, 90 grade
Tragere	da	da	da
Îndoire	da	nu	nu
Umplere	da, 2 variante	nu	nu
Efecte			
Anti-Aliasing	da	da	nu
Colorare	da	nu	da
Transparenţă	da	nu	nu
Perspectivă 3D	da	nu	nu
Ungere	da	da	da
Umbrire	da	da	da
Strălucire	da	da	da
Animaţie culori	da	nu	nu
Degrade	da	da	da
Contrast	nu	da	da
Luminozitate	nu	da	da
Funcţii text			
Seturi de caractere	21 mono, 2 multicolor	35	20 bitmap, 4 outline
Variante	4	3X4	7
Scalabile	nu	da, prin mărirea pixelilor	da, fonturile outline
Degrade	nu	nu	da
Umbrire	nu	nu	da
Fonturi policolore	da, două	nu	nu
Altele			
Lupă	da, de 2 pînă la 16 ori	da, de 4 pînă la 8 ori	da, de 2 pînă la 8 ori
Şabloane de culoare	da	nu	nu
Pagini de rezervă	1	1	niciuna
Paletă liber definibilă	da	da	da
Scanner	nu	nu	da
Pipetă de culoare	da	nu	da
Editor de tranziţie	da	da	nu
Descompunere prin raster	opţional	opţional	nu
Caleidoscop	da, 3 variante	nu	nu
Produs			
Dischete	4X5,25" / 2X3,5"	3X5,25" / 1X3,5"	6X5,25"
Manual	230 pagini	260 pagini	210 pagini
Manual de referinţă	da	nu	nu
Preţ (DM)	399	495	499
Producător	Electronic Arts	Rix Software Inc.	ZSoft Corporation

Formate grafice

Cine doreşte să prelucreze cu ajutorul unui program de grafică o imagine obţinută cu un alt program, se loveşte de probleme. De obicei programele nu vorbesc aceeaşi limbă.

Cine scrie texte cu ajutorul calculatorului, acela doreşte să le şi ilustreze. Din acest motiv, de multe ori, utilizatorul stă în faţa unei probleme: "cum va arăta imaginea grafică, desenată cu programul x, în cadrul procesorului de texte y, sau în cazul programului de publicistică (Desktop Publishing) z ?". Sau: "cum aş putea prelucra cu succes o imagine creată cu un anumit program de grafică, cu propriul program de grafică ?".

Motivul: pentru întîietate se ceartă o sumedenie de programe de grafică, publicistică sau procesare de texte. Pentru a atinge cea mai bună calitate posibilă, în ceea ce priveşte programele, aproape fiecare producător de soft merge pe drumul său propriu, şi lucrează cu formate fişier proprii. Din acest motiv s-a ajuns la o junglă de formate de date în care te poţi rătăci foarte uşor. Pentru a găsi totuşi drumul cel bun aveţi nevoie de cîteva cunoştiinţe de bază şi de programele potrivite. În acest articol încercăm să vă dăm cîteva informaţii necesare.

Să începem cu întrebarea cea mai importantă: Ce este de fapt un format fişier ? Fiecare dintre programele de aplicaţie - fie de desenat, de prelucrare de texte sau bază de date - are cîte o funcţie prin care rezultatul muncii lor (de ex.: scrisori, imagini, listă de adrese) poate fi memorat pentru o durată mai îndelungată. Sub forma unor combinaţii de biţi, deci într-o formă pe care calculatorul o poate percepe, programele inregistrează informaţiile pe dischete sau pe harddisk-uri. Rezultatul este un aşa-numit fişier. Numele lui este conţinut în directorul dischetei sau harddisk-ului. Dacă cerem să vedem conţinutul unui disc, cu comanda DOS DIR, putem observa imediat că numele fişierului este format din două părţi, şi anume: prima din maxim opt caractere (numele sau prefixul), care este stabilită de utilizator, de ex.: "TEST", a doua este terminaţia (extensia sau sufixul), care se pune după punct şi care constă din maxim trei caractere. Extensia este stabilită, de regulă, în mod automat de către program (de ex.: PCX, DRW, etc.).

Conţinutul propriu-zis al fişierului, imaginea deci, poate fi dezvăluită dintr-un fişier grafic, doar cu programul cu care a fost creată. De ex. atunci cînd fişierul este citit de program, interpretat, şi imaginea este afişată pe ecran.

De obicei, încercarea de a citi o imagine, cu un alt program decît cel cu care a fost creată, nu reuşeşte, pentru că metoda cu care de ex. programul de grafică A cifrează informaţia "desenează un cerc cu raza x, avînd centrul în y, de culoarea z", nu este normată şi nu poate fi înţeleasă de alte programe.

În ceea ce priveşte formatele grafice pe PC-uri, există o multitudine babilonică de limbaje. Din acest motiv este greu să poţi avea o privire de ansamblu asupra diferitelor formate. Pentru a vă crea o imagine, vă vom prezenta cele mai importante formate de fişiere utilizate de programele de grafică, pe scurt formatele grafice.

Formatele bitmap

Unele formate sînt legate direct de un program grafic, care scrie şi citeşte exact acel format. Altele au devenit, între timp, un cvasistandard, care este înţeles de mai multe programe diferite. Formatul fişierului poate fi recunoscut după cele trei litere ale extensiei sale.

PCX

Formatul PCX (cu extensia PCX) este unul din formatele cele mai răspîndite pentru programele bitmap şi în acelaşi timp este şi cel mai vechi format de grafică în domeniul PC-urilor. El este foarte strîns legat de firma ZSoft care a elaborat programul de pictură foarte popular "PC Paintbrush", ajuns între timp la versiunea a IV-a. Formatul PCX a traversat unele modificări, deoarece Paintbrush a continuat să fie dezvoltat, mai ales a crescut numărul culorilor ce pot fi reprezentate. Dacă la început puteau fi reprezentate maxim 16 culori (ceea ce corespundea performanţelor plăcilor grafice CGA şi EGA) astăzi s-a ajuns la un număr maxim de 256 de culori dintr-o paletă de 262144, deci adaptat la standardul VGA de astăzi. Cum pot fi alese cele 256 de culori din mulţimea uriaşă de peste 1/4 de milion ? Soluţia este simplă. La începutul fişierului PCX se defineşte o aşa numită paletă, în care sînt înscrise definiţiile valabile ale imaginii, deci care sînt cele 256 de culori folosite. Mărimea şi rezoluţia imaginilor nu este limitată de format, ele depind doar de capacitatea programului de pictură, de memoria disponibilă a calculatorului folosit şi de cartela grafică înglobată. Datorită mărimii lor, fişierele PCX se memorează mai ales sub formă comprimată.

TIFF

Acest format grafic, (prescurtare de la "Tag Image File Format", extensia fişierelor: TIF), a rezultat ca urmare a înţelegerii survenite în anul 1986 între firmele Aldus, Microsoft şi producători de scannere de frunte. TIFF este foarte aproape de a deveni standardul pentru grafica bitmap, mai ales în domeniul profesional şi semiprofesional, atît pe PC cît şi pe Apple.

Spre deosebire de PCX, pentru

TIFF nu există un program anume care să genereze acest format, ca format de bază, existînd în schimb foarte multe programe care pot citi şi scrie acest format. Cu versiunea actuală, 5.0, se pot prelucra imagini alb-negru, grafice cu diferite nuanţe de gri şi imagini cu pînă la 16.7 milioane de culori (în format 24-bit-TIFF).

Structura internă a fişierelor TIFF este fixă şi este cu ceva mai complexă decît cea a fişierelor PCX. Şi în acest caz se foloseşte procedeul de comprimare a datelor. Nu există o limitare în ceea ce priveşte mărimea şi rezoluţia imaginilor.

GIF

Formatul GIF provine din SUA. El a fost proiectat de firma Compuserve din Ohio. Compuserve este unul dintre cei mai mari producători de cutii poştale electronice şi de sisteme informatice ce pot fi folosite prin intermediul unui modem, din lume.

GIF a fost proiectat, mai ales, pentru a pune la dispoziţia sistemelor uzuale PC, Atari ST, Mac, Amiga, un format grafic, care să poată memora datele făcînd economie de spaţiu. De ce chiar programatorii de la Compuserve au pus accent pe acest lucru este clar: timpul necesar transferării graficii prin telefon este lung şi costă bani. Cu cît fişierul este mai mic, cu atît mai repede se poate efectua transferul.

Şi GIF, ca şi PCX, poate folosi 256 de culori dintr-o paletă de 262144 de culori. Pînă la această oră nu cunoaştem decît un program de aplicaţie care ştie scrie şi citi formatul GIF: este vorba despre programul Autodesk Animator. Există, totuşi, cîteva programe freeware, ca de ex. "VPIC.EXE", care poate prezenta şi imagini GIF.

LBM / IFF

Acest format grafic este răspîndit mai ales pe calculatoarele Amiga, numele lui vine de la: "Interchange File Format", extensia fişierelor este IFF. Acest format grafic şi-a făcut intrarea în lumea MS-DOS în momentul în care programul de pictură "Deluxe Paint" a fost transcris de pe Amiga pentru PC. Pe Amiga fişierele aveau extensia LBM. Deci imaginile grafice realizate cu Deluxe Paint pe Amiga sînt compatibile cu cele create de versiunea PC a aceluiaşi program, ele putînd fi încărcate şi prelucrate pe PC. Deoarece calculatoarele din familiile Amiga şi PC, (cu cartelă grafică EGA sau VGA), au rezoluţia ecranului şi raportul lăţime/înălţime diferite, la transferul imaginilor apar deformări, care trebuie corectate ulterior.

Amiga mai posedă şi modul grafic HAM cu 4096 de culori simultane şi cîteva programe de pictură Amiga, de ex. "Photon Paint", folosesc această multitudine de culori. Deoarece nici cartelele VGA moderne, (standard PC), nu pot stăpîni acest mod de lucru, la asemenea grafice pot apare deformări crase de culori, care trebuie corectate ulterior de mînă.

Formatul IFF a fost proiectat nu doar pentru memorarea de grafică ci şi pentru memorarea unor piese muzicale, zgomote, sau text, în acest format standard. În acest caz nu s-a produs nici o extensie înafara domeniului Amiga. Şi IFF/LBM foloseşte metoda de comprimare a datelor.

MSP

Această prescurtare, care coincide şi cu extensia fişierelor, este folosită pentru "Microsoft Paint" şi este numele formatului folosit de programul cu acelaşi nume. Microsoft Paint a fost livrat împreună cu suprafaţa utilizator grafică Microsoft Windows pînă la versiunea 2.11. MSP este un format alb-negru curat, neexistînd opţiunea de culoare. Formatul MSP există în două variante 1.0 şi 2.0. Versiunea 1.0 a fost folosită doar de Microsoft Paint, livrat împreună cu Windows 1.0. Dacă se doreşte folosirea unei astfel de imagini cu versiuni mai noi ale programului, trebuie folosit programul de conversie "CVTPAINT.EXE", care a fost livrat odată cu Windows pînă la versiunea 2.11. Începînd cu versiunea Windows 3.0, Microsoft Paint a fost înlocuit cu o variantă specială a clasicului Paintbrush (proiectat tot de ZSoft).

Această versiune Paintbrush poate utiliza pe lîngă formatul iniţial al casei ZSoft, PCX, şi fişiere MSP vechi, pe care le poate citi, dar nu mai poate scrie în format MSP.

BMP

BMP este prescurtarea pentru Bitmap, extensia BMP, şi este formatul pe care versiunea Paintbrush din Windows 3.0 îl poate scrie şi citi în mod standard. În acest format grafic sînt livrate şi imaginile pe care Windows 3.0 le poate afişa pe fundalul ecranului. Se pare că Microsoft are gînduri mari în legătură cu acest format, pentru următoarele versiuni ale lui Paintbrush. Dacă în Paintbrush alegem meniul "Memorare sub ..", atunci după clic apare cîmpul "opţiuni" cu mai multe posibilităţi de memorare, dintre care se poate alege formatul de memorare: fie format BMP cu 2, cu 16 sau cu 256 de culori, fie format 24-bit, deci cu 16.7 culori. La Windows 3.0 s-au pus deci bazele unei prelucrări cu culori naturale.

IMG

Acest format grafic ar trebui să fie cunoscut posesorilor de calculatoare Schneider de tipul PC1512 şi 1640. Odată cu aceste calculatoare, Digital Research a livrat suprafaţa utilizator grafică "GEM", care conţinea programul "GEM Paint". Acesta folosea formatul IMG (Image), care suporta 16 culori. Cel care deţine multe imagini în format IMG nu trebuie să renunţe la ele, deoarece există nenumărate posibilităţi de a le converti în alte formate mai uzuale.

Targa

Formatul Targa provine din lumea profesională a minicalculatoarelor, a supermini-urilor şi a staţiilor

de lucru. Pe PC aceste fişiere au extensia TGA. Formatul Targa este un format bitmap care poate reprezenta 16.7 milioane de culori şi care este folosit în prelucrarea profesională a imaginilor şi în retuşarea lor. Pe PC există şi programe de grafică vectoriale, cum ar fi Mirage, care pot memora imagini în formatul TGA, deoarece recalcularea unei imagini vectoriale într-un format bitmap nu este mare lucru. Conversia în sens invers nu se poate face, însă, decît cu pierderi mari de calitate.

Astăzi formatele TGA sînt folosite foarte des pentru fixarea imaginilor pe diapozitive deoarece aproape orice studio specializat poate prelucra formate TGA.

SCx

Acesta este formatul de bază al programelor "Colorix EGA Paint" şi "Colorix VGA Paint" de pe PC. Primele două caractere ale extensiei sînt întotdeauna caracterele "SC", al treilea caracter diferind în funcţie de rezoluţie şi de numărul culorilor folosite. La rezoluţia VGA normală, cu 640X480 puncte şi 16 culori, extensia este "SCP". În acest format pot fi memorate imagini în toate rezoluţiile cuprinse între CGA (320X200 puncte cu 4 culori din 16 posibile) şi Super-VGA (1024X768 puncte şi 256 de culori simultan).

MAC

Un alt format bitmap foarte răspîndit pe Apple Macintosh, alături de TIFF, este MAC, formatul de bază în programul de pictură "Macpaint". El poate fi convertit în alte formate cu ajutorul unor utilitare.

Formate vectoriale

În ciuda multitudinii de formate bitmap, pe PC există totuşi două cvasistandarde, şi anume PCX şi TIF, care pot fi citite de aproape toate programele Desktop Publishing.

În cazul formatelor vectoriale este cu totul altfel - nu neapărat mai rău. Toate programele de grafică vectorială au formate proprii de date, care posedă caracteristici specifice.

DRW

DRW este o prescurtare pentru "Drawing" şi este formatul versiunii 3.01 al lui Micrografx Designer. Pe lîngă acesta coexistă şi versiunile 1.0 şi 2.0 care pot fi prelucrate de cea mai nouă versiune a programului. Caracteristici: utilizează curbele Bézier şi poate folosi pînă la 16.7 milioane de culori.

PIC

PIC este un format destul de rar întîlnit în ultima vreme, el poate fi scris de programul "In-A-Vision" al casei de soft Micrografx. Programul In-A-Vision a apărut în iunie 1985 şi a fost una din primele aplicaţii pentru Windows şi unul dintre primele programe de desenare orientate pe obiecte de pe PC.

GRF

GRF este formatul programului Charisma, un program pentru grafică economică, produs de Micrografx. Acest format stochează atît informaţiile grafice cît şi informaţii despre datele dintr-un fişier propriu. În rest se aseamănă cu formatul DRW. Charisma poate memora o imagine şi în format DRW.

XLC

XLC este prescurtarea pentru "Excel Chart". Acesta este formatul în care programul de calcul tabelar Excel, al lui Microsoft, îşi salvează graficele.

Lotus PIC

Lotus PIC este formatul folosit de programul de calcul tabelar Lotus 1-2-3 pentru memorarea graficelor şi imaginilor produse.

Pict I & II

Acesta este un format vectorial foarte răspîndit pe Apple Macintosh. Este folosit, de ex., de programul de desen "MacDraw".

GEM

Acest format, cu extensia GEM, este folosit de suprafaţa utilizator grafică cu acelaşi nume pentru memorarea graficelor vectoriale. Sub acest nume există două formate: în primul rînd formatul simplu al lui "GEM Draw", şi în al doilea rînd formatul lui "GEM Artline 2.0", care posedă, faţă de Draw, cîteva caracteristici în plus, de ex.: tranziţia culorilor şi aşa-numitele curbe Bézier.

CDR

CDR este formatul de bază folosit de programul grafic "Corel Draw".

CED

CED este formatul de bază folosit de programul grafic "Arts & Letters".

DXF, SLD, ADI

Formate de export/import pe care programul "Autocad" al firmei Autodesk le foloseşte pentru schimbul de date cu alte programe.

WMF

WMF este prescurtarea pentru "Windows Metafile", un format vectorial folosit de Windows pentru schimbul de date între programe.

CGM

CGM este prescurtarea pentru "Computer Draphics Metafile" şi un format de fişier foarte des folosit pentru transferul de date între diferite programe de grafică şi între diferite sisteme de calcul. CGM nu este formatul de bază al unui anumit program grafic, ci apare ca opţiune de export/import a unor programe.

(continuare în pag. 23)

Babilonia limbajelor de programare

Unul ştie doar C, altul are încredere doar în dialectele de Basic, un al treilea fără compilatorul Pascal ar fi pierdut, şi fiecare crede că doar limbajul "lui" îţi poate da mintea cea de pe urmă. Această introducere îşi propune să arate de ce greşesc cu toţii.

Cel care consideră că unul sau altul dintre limbajele de programare reprezintă singura şi cea mai bună soluţie, acela se înşeală. Legea inerţiei este responsabilă de inexistenţa unui limbaj universal pentru toată lumea. Căci cine nu este obligat de profesie, sau de specializare să treacă de la un limbaj, învăţat cu greu, la un alt limbaj, acela nici n-o va face. Nici un african nu va învăţa chineza, doar pentru faptul că mai mult de un miliard de pămînteni o folosesc.

În plus, în ceea ce priveşte limbajele evoluate, nu se observă diferenţe semnificative în ceea ce priveşte performanţele compilatoarelor. Nu cunoaştem nici o problemă care să poată fi rezolvată doar în C, doar în Basic sau doar în Pascal.

Strategi iscusiţi ai reclamei şi fetişişti ai compilatoarelor reuşesc mai întotdeauna să demonstreze, cu ajutorul unor exemple bine alese, că un limbaj anume este cel mai bun. Ei n-ar trebui crezuţi pe cuvînt, deoarece un adevărat profesionist ar trebui să stăpînească cel puţin trei limbaje evoluate şi două asambloare. Un motiv suficient pentru această situaţie l-ar putea constitui, de ex., posibilitatea ca lucrînd într-o echipă să se ia decizia că o anumită aplicaţie trebuie scrisă într-un anumit limbaj de programare, de ex. C, sau, de ex., se poate întîmpla ca o aplicaţie deja scrisă, să zicem în Pascal, să trebuiască dezvoltată folosind acelaşi limbaj de programare. Mediul de dezvoltare API, (Application Programm Interface - o colecţie de rutine, gata scrise, pentru utilizarea caracteristicilor sistemului), oferit de sistemul de operare pe care se lucrează, joacă un rol hotărîtor în alegerea limbajului de programare ce urmează a fi folosit.

Sistemul de operare al calculatoarelor Apple Macintosh, de ex., a fost scris în Pascal, toată documentaţia şi toate exemplele puse la dispoziţie sînt concepute în Pascal. Deci cel mai eficient limbaj de programare a sistemului este limbajul Pascal. Altfel se prezintă situaţia în cazul suprafeţei utilizator PC Windows 3.0. Ea a fost programată în limbajul C, în C fiind scrise şi exemplele, bibliotecile şi fişierele antet (header) ale sistemului. Un programator fără cunoştiinţe de C se descurcă greu în acest caz.

API-urile scrise în C domină net: MS Windows, OS/2, GEM, Atari ST, Amiga, X-Windows şi întreaga lume Unix sînt scrise în C. La Apple, Pascal este încă puternic, dar C rămîne limbajul principal de programare a sistemelor. Tot aici îşi găsesc principalul domeniu de aplicabilitate şi asambloarele. Ele sînt folosite atunci cînd trebuie rezolvată o problemă critică în mod optim, sau cînd trebuie făcută o programare apropiată de hardware. De aceea compilatoarele bune permit inserarea unor secvenţe de instrucţiuni scrise în limbaj de asamblare direct în textul sursă, iar limbajele de asamblare moderne, cum ar fi MASM 5.1 al lui Microsoft, oferă interfeţe confortabile spre C.

Concluzia este că programele pentru un API nu pot fi scrise în Basic sau Pascal. Într-adevăr, fără un efort considerabil, nu pot fi scrise aplicaţii Windows adevărate în Turbo Pascal. Totuşi pot fi scrise programe care să arate asemănător. Cu toate că a durat destul de mult, între timp şi Borland sprijină, cu Turbo Pascal 6.0, programarea ferestrelor, a meniurilor Pull-Down şi a căsuţelor de dialog, cît şi deservirea cu mouse-ul. Nici Microsoft, inventatorul acestei suprafeţe SAA, n-a fost mai iute şi abia odată cu versiunea 7.0 a Basic-ului a livrat şi modulele necesare.

În concluzie: cine doreşte să programeze un sistem cu un API scris în C, în această categorie fiind în primul rînd OS/2 şi Windows 3.0, acela nu poate trece pe moment peste C. La Amiga, Atari ST şi Macintosh s-au dezvoltat între-timp versiuni Basic, C şi Modula care sprijină un acces direct la API. În mod cert cauza acestei stări de fapt o constituie faptul că la aceste calculatoare funcţiile componente fac parte din sistemul de operare, în timp ce Windows nu este decît o extensie a sistemului de operare DOS, pe care o poţi avea sau nu.

După cum am mai precizat, versiunile Basic moderne pot face acelaşi lucru ca şi în Pascal, dacă luăm în considerare doar ceea ce se poate face cu aceste sisteme. Cum se poate face aceasta este o altă întrebare. Din punctul de vedere al unui programator Pascal, o instrucţiune de genul "DIM i AS INTEGER" în loc de "i integer", într-o declaraţie de articol, în care cuvîntul "record" nici nu apare, sună ciudat. Pentru un programator care a crescut cu Basic şi cunoaşte această sintaxă, care în urmă cu doi ani nici nu exista, totul pare cît se poate de normal. El nu va găsi nici un motiv de a trece la Pascal. Dar cu ce limbaj să începi ?

Diferenţa dintre cele două limbaje constă în faptul că declaraţia variabilelor în Basic este facultativă, în timp ce în Pascal este obligatorie, o conversie de tip se face automat în Basic, în timp ce în Pascal trebuie făcută explicit.

Altfel spus: în Pascal papa Wirth ia programatorul de mînă şi îl trece

strada, în Basic are voie să treacă singur, iar în C poate trece chiar şi atunci cînd semaforul este pe roşu. Deci un începător n-ar trebui să înceapă cu C, mai rămîn Basic şi Pascal.

Un avantaj la Basic era ritmul interpretorului, introducerea instrucţiunilor, urmată de RUN, ducea la apariţia instantanee a răspunsului pe ecran, în timp ce la Pascal trebuia trecut prin lanţul editor-compilator-linkeditor-test. Cu ajutorul unor medii de dezvoltare integrate şi a unor compilatoare foarte rapide, (un principiu care a adus faimă lui Turbo Pascal), timpul de execuţie a ajuns între timp acelaşi, dacă facem abstracţie de programele foarte mici. În cazul lor Basic se află în mod cert în frunte. Programatorii folosesc acest avantaj atunci cînd doresc un transfer rapid de fişiere între două calculatoare, cu un program de cinci linii. În acest caz MS-Basic cu al său OPENCOM nu poate fi întrecut de nimeni, şi mai are încă un avantaj: specificarea parametrilor de interfaţă se face şi la Amiga şi la Macintosh şi la PC la fel. În schimb la Pascal, şi chiar la C, pentru acest lucru trebuie cunoscute cîteva detalii despre interiorul sistemului de operare şi/sau despre funcţiile de bibliotecă. Pînă le-am afla, sau am căuta în manual, programul Basic ar rula deja.

Exemplul ce urmează va arăta un alt avantaj al Basic-ului: un programator a scris un program destul de complex în MS-Basic pe Macintosh. După cîţiva ani îl transferă pe Amiga, programul va porni instantaneu. Orice program Basic PC, fără instrucţiunile PEEK şi POKE, funcţionează şi pe un Macintosh, dar nici un program Pascal PC, scris în Turbo Pascal, nu va funcţiona pe un Mac sub Turbo Pascalul de acolo.

Se poate spune că MS-Basic, (sau GW Basic), a devenit un standard pentru mai multe calculatoare, în timp ce la Turbo Pascal, limbajul cel mai standardizat la origine, diferenţele între dialecte sînt sensibile.

Şi o ultimă deosebire: datele şi programele pot fi structurate foarte bine şi în Basic-ul modern, dar acest lucru se face facultativ, în timp ce la Pascal această structură formează conceptul limbajului şi apare în mod natural. De aceea programele mari, mai ales cele orientate obiect, se pot formula mult mai elegant în Pascal.

Începătorul are deci două posibilităţi: fie începe cu Basic, caz în care la început nu va trebui să facă eforturi prea mari, dar apoi trebuie să treacă singur la învăţarea tehnicilor moderne de programare, fie începe cu Pascal, va studia mai mult limbajul decît soluţiile problemelor, dar îi va fi mai uşor mai tîrziu. În orice caz, dacă primul limbaj învăţat este Pascal, al doilea se învaţă mai uşor.

Decizia de a utiliza unul sau altul dintre limbaje va trebui să depindă de scopul pentru care programatorul doreşte să utilizeze compilatorul. Tabelul anexat poate servi pentru o orientare corectă. S-au acordat note între 0 şi 3, 0 pentru cea mai rea apreciere şi 3 pentru cea mai bună, pentru fiecare criteriu în parte.

Criteriu	Asamblor	C	Basic	Pascal
Efort de învăţare redus	0	1	3	2
Probleme puţine la început	0	1	3	2
Deschizător de drumuri	0	2	1	3
Programe mici	0	1	3	2
Obţinerea rapidă a unei soluţii	0	1	3	2
Programe mari	0	3	1	2
Proiecte mari (echipă)	0	3	0	2
Structuri complexe de date	0	2	1	3
Structuri de program complexe	0	2	1	3
Apropiat de sistem	3	2	0	1
Apropiat de hardware	3	2	0	1
Soluţii cu timp de rezolvare critic	3	2	0	1
Cod compact	3	2	0	1
Acces API	1	3	0	2
Ofertă module/biblioteci	0	3	1	2
Utiiitare suplimentare	0	3	1	2
Aplicaţii OOP	0	2	1	3
Facilităţi OOP	0	3	1	2
Portabilitate	0	3	2	1
Siguranţă în viitor	0	3	2	1

Tabelul arată care limbaj pentru care scop este mai indicat

Şi-au mai rămas doar 10

Oferta de noi şi noi medii de programare este de necuprins. Dacă stai însă să alegi făina de tărîţe, atunci mai rămîn doar cîteva. Vă prezentăm în continuare cele mai importante 10 compilatoare.

În căutarea celor mai bune compilatoare te loveşti inevitabil de două nume: Microsoft şi Borland. Aceste două firme sînt, de multă vreme, cei mai mari, şi aproape şi singurii, ofertanţi de medii de dezvoltare software profesionale. Cu multă pricepere au reuşit să-şi împartă piaţa. Ca urmare a acestui cvasi-dublu monopol există o apropiere între structura şi puterea compilatoarelor oferite.

Între-timp a devenit neimportant ce compilator este folosit: Basic, C, Pascal sau chiar Assembler. Puterea limbajelor de nivel înalt a devenit atît de mare încît nu mai există probleme care să poată fi rezolvate doar într-un anumit limbaj. Assembler-ul îşi păstrează şi pe mai departe rolul de unealtă a specialiştilor.

Şi pentru a face puţină lumină în lumea compilatoarelor, pe următoarele pagini, vă prezentăm cele mai importante 10 compilatoare. De remarcat este faptul că

prezentarea va fi amestecată, deci vor fi puse în relaţie compilatoare de diferite categorii. Prezentarea nu urmăreşte să stabilească care compilator de Basic este cel mai bun compilator de Basic, ci care este cea mai bună unealtă de dezvoltare aplicaţii ?

Quick-C cu Quick-Assembler 2.5

Quick-C cu Quick-Assembler este o combinaţie dintre un compilator C şi un asamblor care funcţionează sub aceeaşi suprafaţă utilizator. Pachetul este compatibil cu MS-C 6.0, MASM 5.1 şi cu Debugger-ul din trusa de scule (Programmer's Workbench) oferită de MS-C 6.0. Diferenţa de bază faţă de versiunea profesională este că pot fi create programe care să ruleze doar în Real-Mode şi nu şi pe OS/2.

Mediul integrat de dezvoltare dispune de meniuri pull-down şi de ferestre în standardul SAA. Din interiorul editorului pot fi scrise programe, compilate, linkeditate, lansate în execuţie şi depanate.

La acestea se adaugă un sistem de help-uri contextuale, de primă mînă, care conţin practic toate manualele, un curs de C şi o multitudine de exemple. În ceea ce priveşte timpul turn-around, timpul de aşteptare în cazul asamblorului este practic nul. Compilatorul C poate avea nevoie de cîteva secunde deoarece scrie chiar la început pe disc. Dacă se dispune de un RAM-disc, 4000 de instrucţiuni sînt compilate în mai puţin de o secundă, ceea ce duce la concluzia că timpii de compilare depind doar de timpul de acces la harddisk. Depanatorul dispune de funcţii tracing, backtracing, undo şi permite fixarea de puncte de întrerupere. Conţinutul variabilelor poate fi examinat şi modificat, iar în cazul asamblorului acelaşi lucru este valabil şi pentru conţinutul registrelor.

Pentru Quick-C există biblioteci bogate, în standard ANSI, şi există şi biblioteci speciale, de ex. de grafică. Asamblorul este super. În loc de bucla obişnuită editor- asamblor- linkeditor- EXE2BIN- debugger- editor, se rămîne în editor, o apăsare de tastă este suficientă pentru tot ciclul.

În furnitura de livrare, a versiunii 2.5, există trei manuale groase şi un caiet numit "Up and Runing". Manualele "Programmer's Guide" şi "C for Yourself" sînt manuale de prima mînă pentru începători.

Quick-Basic 4.5

Şi Quick-Basic poate fi descris foarte simplu. Este Quick-C, cu singura deosebire că textul sursă pretins de compilator trebuie să fie în Basic. Acest Basic are cîteva caracteristici, cu care, (din păcate), Pascal şi C nu se pot lăuda. Astfel editorul dispune de un analizor de sintaxă care evaluează fiecare instrucţiune în momentul introducerii, ceea ce face ca timpul turn-around să fie practic nul. Motivul: fiecare instrucţiune este tradusă, imediat după introducere, într-un pseudocod (Threaded P-Code). Abia după ce se consideră că totul poate rămîne aşa cum s-a introdus, se apelează (restul) compilatorului care atinge o viteză de 150.000 de instrucţiuni pe minut.

Datorită diferitelor structuri de control, inclusiv selector CASE, se pot folosi tehnici de programare care să nu stea cu nimic în urma celor din Pascal. Sînt posibile chiar şi definiţii de articole şi de tipuri de date Pascal. Cu ajutorul instrucţiunii DIM extinse pot fi făcute şi declaraţii de variabile.

FUNCTION corespunde în modul de scriere şi în efect cu funcţia corespondentă din Pascal, dacă se utilizează şi cuvîntul cheie STATIC atunci se cumulează şi avantajele lui C. Pe lîngă funcţii mai există şi proceduri care cu excepţia cuvintelor cheie SUB şi END SUB nu pot fi deosebite de cele din Pascal. Parametrii pot fi transmişi prin valoare sau prin referinţă (adresă).

Ergo: Quick Basic 4.5 este un compilator Basic modern, foarte puternic.

Quick-Pascal 1.0

Ultimul reprezentant al familiei Quick este Quick-Pascal. El se înscrie pe aceeaşi linie ca şi precedesoarele, dar pretinde text sursă în Pascal. Limbajul în sine este aproape o copie a lui Turbo Pascal 5.0, fără overlay-uri, dar cu o extensie pentru programarea orientată obiect. Există puţine cuvinte cheie pentru Pascal orientat obiect: cel mai important este OBJECT care permite definirea de obiecte şi clase. Mai sînt şi cuvintele cheie OVERRIDE şi INHERITED. Aceasta este totul, restul este Turbo Pascal.

Quick Pascal şi Turbo Pascal 6.0, în comparaţie cu sistemele OOP de pe Macintosh şi cu MS-Windows, au unele lipsuri, deoarece acolo tehnicile OOP sînt integrate deja în sistemul de operare, respectiv în extensia acestuia. Totuşi Microsoft este pe fir, şi chiar acest lucru vrea să-l modifice şi încă în DOS.

OOP pe baza sistemului poate avea consecinţe cu bătaie lungă. Metoda unui obiect s-ar putea numi atunci: dă-ţi drumul atunci cînd se întîmplă un anumit eveniment în sistem. Aceste informaţii despre evenimente ar putea fi declanşate, ca şi sub Windows, de mouse, de tastatură sau de ceas.

Combinaţia control prin evenimente şi OOP este tehnica de programare ideală. S-ar defini simplu un obiect, de ex. un icon (imagine grafică care simbolizează o anumită acţiune), şi i s-ar defini o metodă de genul: porneşte atunci cînd se dă un clic pe mouse deasupra ta. Următorul obiect ar putea aştepta apăsarea tastei F1, şi al treilea (ceasul) ar porni atunci cînd ar primi o anumită informaţie de la generatorul de tact. Ar fi treaba sistemului de operare să semnaleze obiectelor producerea unor astfel de evenimente. Avantajul: obiectele ar fi pe deplin independente unul

de celălalt, dar ar lucra totuşi împreună, atunci cînd ar fi nevoie. S-ar putea adăuga un nou obiect, şi deci noi facilităţi, într-un program, fără a fi nevoie să se modifice nici măcar o singură literă în programul existent.

Nu ne rămîne decît să sperăm că Microsoft va avansa rapid pe acest drum pe care a pornit.

Turbo Pascal 6.0

Turbo Pascal pînă în versiunea 5.0 este în general cunoscut în cercurile programatorilor. Versiunea 5.5 a adus în plus programarea orientată obiect (OOP), care a fost desăvîrşită în versiunea 6.0, şi care permite scrierea unor programe cu totul noi. Pentru instalarea sistemului utilizatorul nu trebuie decît să tastèze: INSTALL. Un harddisk este obligatoriu, un mouse recomandabil.

Mediul integrat de dezvoltare oferă meniuri pull-down şi ferestre aproape în standard SAA. Din editorul Multi-Windows/Multi-File programele pot fi compilate, linkeditate, lansate în execuţie şi depanate.

Timpul turn-around este foarte scurt. Dintr-un program de mai bine de 1400 de instrucţiuni plus trei unităţi, pe un 386, după cîteva zecimi de secundă, se obţine un fişier EXE în memorie, care în mai puţin de o secundă este transferat pe disc.

Dacă memoria disponibilă nu este suficientă - cca. 270 KBytes - se poate folosi varianta TPC, sau extensia profesională TPCX, care lucrează în mod protejat şi în memoria extinsă.

Helpul contextual permite un help în genul celui de la compilatoarele Quick. Lăudabilă este noua fereastră "Registers" care permite examinarea conţinutului registrelor unităţii centrale. Aceasta se potriveşte cu noua instrucţiune ASM care permite inserarea în programul sursă a unor secvenţe scrise direct în limbaj de asamblare.

Cu Turbo Pascal 6.0 se livrează şi "Vision", un cadru orientat obiect pentru programe comandate de evenimente. În opt unităţi sînt oferite obiecte pentru manevrarea de ferestre suprapuse, pentru scalarea ferestrelor, controlul mouseului, crearea de meniuri pull-down, boxe de dialog, manipularea evenimentelor (tastatură, mouse) şi o paletă bogată de controale.

Turbo Pascal 6.0 se dovedeşte consecvent cu el însuşi în tratarea modalităţilor OOP introduse odată cu versiunea 5.5. Rezultatul este un mediu de dezvoltare, orientat grafic, reuşit cu multe obiecte şi metode utile.

Dar despre Turbo Pascal 6.0 şi despre Turbo Vision vă vom spune mai multe într-un număr viitor.

MS-Basic 7.1 - PDS

O analiză a pieţii din SUA făcută de Microsoft a dus la concluzia că peste 50% dintre cumpărătorii de Quick-Basic sînt programatori profesionişti şi că 62% dintre aceştia scriu aplicaţii comerciale. S-a stabilit de asemenea că cerinţele crescînde ale portabilităţii nu pot fi satisfăcute dacă nu sînt respectate următoarele condiţii: Basic trebuie să fie tot la fel de rapid ca şi Turbo Pascal, să susţină algoritmi puternici de tratare a bazelor de date, overlay-uri, OS/2 şi EMS, şi să ofere toate metodele moderne de structurare a programelor şi datelor. Pe de altă parte se doreau şi obişnuitele utilitare de la Make şi bibliotecar (Library Manager), pînă la depanatorul simbolic comandabil prin mouse, şi se dorea să se rămînă în mediul obişnuit al lui Quick-Basic.

Instalarea se face de pe 12 dischete şi după instalare, în funcţie de dorinţa utilizatorului, pe harddisk va fi ocupat un spaţiu de 8 pînă la 14 MBytes; astea da dimensiuni !

Basic 7 este compilatorul profesional BC 6 extins cu o multitudine de împrumuturi. Astfel funcţiile de calcul financiar şi de calcul al datei, sau puternicele formatări numerice, provin de la Excel, depanatorul de la Codeview, şi suprafaţa utilizator provine dintr-o încrucişare între Quick-Basic 4.5 şi Workbench C 6.0.

Cu ajutorul lui pot fi scrise programe pînă la o dimensiune de 16 MBytes, cu overlay-uri şi recunoaştere automată EMS. Dacă este disponibilă o memorie suficientă, overlay-urile sînt încărcate direct, altfel se face o swapare via harddisk.

Impresionantă este tratarea integrată a bazelor de date ISAM. Ea este activată cu instrucţiunea "OPEN FOR ISAM". Şi încă o noutate este demnă de remarcat: acum programatorii îşi pot dota aplicaţiile cu suprafeţe grafice, modulele necesare aflîndu-se în furnitura de livrare.

Ar mai rămîne de remarcat doar că în medie se obţine un cod cu 68% mai mic decît cel obţinut cu Quick-Basic 4.5, şi că în 37 de teste matematice s-a dovedit cu 225% mai rapid.

Assembler MASM 5.1

MASM face parte din categoria celor mai cunoscute compilatoare, sînt mulţi cei care dispun de o versiune oarecare de MASM, dar sînt puţini cei care cunosc facilităţile noii versiuni 5.1.

În loc să vină cu sfatul lapidar: "luaţi un editor ASCII ..." MASM 5.1 dispune de un editor propriu: "M". Acesta este liber configurabil, Multi-Window/Multi-File şi totodată un mediu integrat. Din cadrul editorului se poate asambla şi linkedita, iar la apariţia unei erori cursorul este poziţionat automat în linia cu pricina. Suprafaţa utilizator orientată tastatură necesită mai întîi o acomodare, dar apoi se dovedeşte a fi extrem de puternică. Confortul obişnuit SAA apare din nou abia în

depanator, care provine dintr-o nouă versiune de la Codeview.

Instalarea de pe cele cinci dischete se rezumă la a tasta: "SETUP" şi a răspunde la întrebarea: "OS/2 sau DOS ?". De fapt sînt livrate două versiuni ale asamblorului. A fost testată doar versiunea DOS căci a programa sub OS/2 în limbaj de asamblare este o artă în sine.

Asamblorul însuşi străluceşte prin modalităţile puternice de macrouri, inclusiv recursive, printr-o mulţime uriaşă de operatori pentru expresii complexe şi prin tratarea procesorului 80386 şi a coprocesorului 80387. O sintaxă strictă şi o verificare strictă de tip ajută la evitarea scamelor."Typos" se utilizează mai rar deoarece există noi directive de segmentare pentru toate apelurile la DOS şi BIOS, aşazise pentru mixare ("mixed"). Ultima permite, alături de alte directive, legătura cu limbajele de nivel înalt. Rutine asamblor pot fi scrise şi ca funcţii C sau Pascal. Dar spre deosebire de codul "inline" (inserat direct în sursă), rutinele scrise în limbaje de nivel înalt vor constitui module, sau biblioteci.

De anunţat ar mai fi utilitarele: BIND, IMPLIB, ILINK şi EXEHDR, care permit conlucrarea cu OS/2 şi Windows.

Turbo-Assembler 2.0

TASM este coresponentul Borland la MASM, versiunile 4.x, şi cu aceasta încă un produs al noii linii cu care cei doi giganţi soft "se ceartă" pentru piaţa atît de frumos împărţită înainte. Ca urmare TASM este compatibil cu MASM tot la fel cum şi Quick-Pascal este compatibil cu Turbo-Pascal, totuşi în cîteva mărunţişuri nu se potrivesc.

TASM solo ne dă, în manual, măreţul mesaj: "luaţi un editor ASCII şi ...", deoarece în furnitura de livrare nu este conţinut şi un editor. De fapt nici nu este necesar, deoarece cu toate că sistemul este altfel complet, el nu-şi poate dovedi întreaga putere decît împreună cu un mediu integrat de dezvoltare Borland (IDE - Integrated Development Environment).

Asamblorul străluceşte, ca şi înaintaşii săi, printr-un macroprocesor puternic, însă nu este actualizat cu toate facilităţile oferite de MASM 5.1, şi din acest motiv nu posedă întreaga gamă de operatori pentru expresii complexe. Dar chiar şi facilităţile oferite de MASM 4.0 erau atît de bogate, încît nu erau folosite la întreaga capacitate decît rar.

Şi TASM sprijină întreaga gamă de procesoare pînă la 80386 şi 80387. El poate lucra şi "mixat", manualul descriind pe larg interfeţele spre celelalte limbaje de programare din familia Turbo. OS/2 în limbaj de asamblare este o muncă grea şi nici programarea Windows nu este aşa de simplă.

Microsoft-C 6.0 - PDS

C 6.0 - PDS - PWB, ceea ce vine de la Professional Development System şi Programmer's Workbench, este un mediu de dezvoltare C pentru DOS, OS/2 şi Windows 3.0. Instalarea se face de pe şapte dischete de 1.2 MBytes şi duce la ocuparea a 15 MBytes pe harddisk.

La prima vedere mediul de lucru se aseamănă cu mediul lui Quick-C, totuşi el oferă mai mult. Este oferit un editor multi-fereastră/ multi-fişier, înregistrator de macrouri, fişiere make complexe şi toate acestea la o simplă apăsare pe tasta mouse-ului.

Componentul Source-Browser permite examinarea programelor complexe. Sistemul de help-uri, după principiul hipertext, este excelent şi conţine informaţiile din cinci manuale, dar şi ocupă 1.9 MBytes pe harddisk.

În ceea ce priveşte timpul turnaround: la o compilare şi linkeditare incrementală, pe un 386, se atinge viteza de 80.000 de linii/minut. Cu toate switch-urile, pînă la optimizarea funcţiilor şi la o optimizare pragmatică, programele mari au nevoie de cîteva minute. Timpul de execuţie, al codului, bate ulterior însă atît timpii obţinuţi cu Quick-C cît şi pe cei obţinuţi cu Turbo-C, cu o evidenţă dramatică. Inserînd inline cod în limbaj de asamblare tempoul poate fi sporit şi mai mult.

Noul Codeview 3.0 poate depana acum şi programe care constau din mai multe procese, aşa-zisele programe cu DLL, şi aceasta cu o editare multi-fereastră. Pe scurt MS-C 6.0 poate fi descris astfel: mediu de lucru optim, cu unelte care se pot folosi foarte comod, sistem deschis (configurabil şi extensibil), cu tehnologii noi de optimizare a codului, cu noi caracteristici limbaj, suport perfect pentru Windows 3.0 şi OS/2 şi cu un depanator de generaţia a treia.

MS-C 6.0 este un sistem pentru profesioniştii care trebuie să elaboreze aplicaţii complexe pentru DOS, Windows şi OS/2.

Turbo C++, Prof. Version 1.0

Dacă Turbo C++ ar permite programarea sistemelor MS-Windows şi OS/2, ar putea fi pus în aceeaşi linie cu MS-C 6.0. Versiunea pentru Windows va apare în curînd pe piaţă, pentru OS/2, Borland crede că încă nu există piaţă.

Altfel este respectată compatibilitatea cu standardul ANSI, de care Borland se ţine cu fermitate. Cei doi de plus suplimentari urmează directivele AT&T.

Instalarea se face de pe 15 dischete de 360 KBytes, şi necesită un spaţiu pe harddisk de 5 MBytes.

Mediul de dezvoltare integrat oferă meniuri pull-down şi ferestre (aproape) în standardul SAA. Mediul de lucru aminteşte de cel de la Turbo Pascal (ca să fim corecţi situaţia este inversă deoarece mediul lui Turbo C++ a fost primul).

Tabel comparativ

Nume	Quick-C cu Assembler 2.5	Quick-Basic 4.5	Quick-Pascal 1.0	Turbo-Pascal 6.0	MS-Basic 7.1-PDS	Assembler MASM 5.1	Turbo-Ass. 2.0 cu Tools	MS-C 6.0-PCS	Turbo C++ Prof.. V.1.0	Zortech C++ - Dev.-Ed. 2.1
Preţ (DM)	390	350	350	520	1190	350	400	1390	600	350
Date tehnice										
Necesită harddisk	da	da	da	da	da	nu	nu	da	da	da
Spaţiu harddisk necesitat (MBytes)	2.5	2	2	5	8, resp. 14	1.5	1.5	15	5	6.5
Suprafaţă SAA	da	da	da	apropiată	da	nu	apropiată	da	apropiată	apropiată
Editor multi-fereastră/multi-fişier	2 ferestre	nu	da	da	da	da	da	da	da	da
Suportă mouse	da	da	da	da	da	nu	da	da	da	nu
Utilizează EMS	nu	nu	nu	da	da	nu	da	da	da	da
Help on-line	da	da	da	da	da	da	da	da	da	aproape
Utilitare de dezvoltare										
Debugger	da	da	da	da	da	nu	da	da	da	da
Profiler	nu	nu	nu	nu	nu	nu	da	nu	da	nu
Source-Browser	nu	nu	nu	nu	da	nu	nu	da	nu	nu
Inline-Assembler	da	nu	nu	da	nu	nu	nu	da	da	da
Macroassembler	nu	nu	nu	nu	nu	da	da	nu	da	nu
Overlay-Manager transparent	nu	nu	nu	nu	da	nu	nu	nu	da	da
Tehnici de programare										
OOP	nu	nu	da	da	nu	nu	nu	nu	da	da
++	nu	nu	nu	nu	nu	nu	nu	nu	da	da
Sistem aprijinit										
DOS	da	da	da	da	da	da	da	da	da	da
OS/2	nu	nu	nu	nu	da	da	nu	da	nu	da
Windows 3.0	nu	nu	nu	nu	nu	da	nu	da	nu	da

În orice caz din mediul IDE pot fi compilate programe, linkeditate, lansate în execuţie şi depanate.

În ceea ce priveşte timpul turn-around: compilatorul este cu ceva mai lent decît cel al lui MS-C. Calitatea codului este foarte bună şi poate ţine pasul cu cea oferită de MS-C 6.0. Depanatorul este foarte bun, avînd întreaga funcţionalitate a lui Codeview, dar fiind extins şi pentru OOP. Lăudabil este preţul de numai 600 DM, sau chiar 250 DM (fără extensia OOP).

Turbo C++ este un sistem pentru profesionişti care poate da clasă lui MS-C 6.0 nu numai în ceea ce priveşte preţul.

Zortech C++, Dev. Edition 2.1

Zortech C++ ar putea fi aproape echivalat cu MS-C 6.0. O singură diferenţă: colecţia de unelte este mai puţin puternică, dar în schimb Zortech C îi oferă pe cei doi de plus. Compatibilitatea se păstrează şi în ceea ce priveşte componentele Codeview şi MASM.

La prima vedere bancul de lucru pare aidoma celui oferit de Turbo C++, totuşi la o privire mai atentă se poate observa că oferă mult mai puţin decît oferă Borland. În ceea ce priveşte timpul turn-around: compilatorul în sine este mult mai rapid decît MS-C. Un apel la bancul de lucru şi returul, pentru un mini-program durează o secundă. Pînă cînd MS-C face acelaşi lucru pot trece pînă la 40 de secunde.

Calitatea codului este foarte bună, un motiv pentru acest lucru: aproape toate sursele din biblioteca oferită sînt scrise în limbaj de asamblare.

(Computer Persönlich 1/91,
Peter Wollsclaeger)

Operatori

aritmetici:	Basic:	+	-	*	/	Ş		
	Pascal:	+	-	*	/	DIV	MOD	
	C:	+	-	*	/	%	++	--
logici:	Basic:	AND	OR	NOT				
	Pascal:	AND	OR	NOT				
	C:	&&	ăă	!				
pe biţi:	Basic:	AND	OR	NOT	XOR			
	Pascal:	AND	OR	NOT	XOR	SHL	SHR	
	C:	&	ă	Ş	!	ş	<	>
atribuire:	Basic:	=						
	Pascal:	:=						
	C:	=	+=	-=	*=	/=	%=	&=
		Ş=	ă=	<=	>=			

Numărul operatorilor diferiţi în C este neatins

Pe drumul spre o tehnică "inteligentă ?"

REŢELE NEURONALE

Partea I

Sistemul nervos uman prelucrează mai multe miliarde de informaţii in modul său propriu. Acest mod de lucru al biologicului şi transpunerea sa în sisteme tehnice este astazi parte integrantă a domeniului Inteligenţei Artificiale. Stadiul şi posibilele influenţe ale acestor cercetări vrem să vi le prezentăm în acest curs structurat în patru părti.

Cuprinsul cursului:

- **Partea 1:** Privire de ansamblu asupra temei
- **Partea 2:** Teoria propagării înapoi
- **Partea 3:** Sisteme expert şi asociativitate
- **Partea 4:** Recunoaşterea scrisului

"Science fiction"-ul a fost perceput de cele mai multe ori ca un amestec de ipoteze utopice şi irealiste avansate de anumiţi scriitori.Dar uneori aceste ipoteze nu sînt altceva decît posibilităţi ştiinţifice reale nerezolvate încă. Că aceste ipoteze pot rămîne încă mult timp inaccesibile este de la sine înţeles. Şi totuşi,ipoteza calculatoarelor "inteligente",visul sau uneori coşmarul,unor scriitori de "science fiction" este deja prezent în unele încercări. A stoca informaţiile deja existente, a le analiza, a reacţiona pe baza lor aşa cum reacţionează un organism uman, toate acestea au fost pînă acum mari probleme nerezolvate ale cercetării în neuroinformatică. În cursul nostru ne vom ocupa de stadiul actual al acestor cercetări şi, mai înainte de toate, vă vom sugera cît mai multe aplicaţii practice ale lor.

Tînărul domeniu de cercetare al Neuroinformaticii se ocupă deci cu transpunerea modului de prelucrare a informaţiei folosit de sistemul nervos uman în reţele neuronale artificiale folosite în tehnică. Ca şi în cercetările generale ale Inteligenţei Artificiale, cu ajutorul reţelelor neuronale se încearcă dezvoltarea unor sisteme cu performanţe "inteligente" cum ar fi recunoaşterea imaginilor (recunoaşterea unor mostre de obiecte, dar şi a unor sunete şi cuvinte articulate), memoria asociativă a cunoştinţelor sau conducerea mişcărilor unor roboţi. Domeniile de aplicare ale acestor sisteme sînt foarte variate, de exemplu în medicină (comanda protezelor, sisteme de diagnoză, analiza radiografiilor) sau tehnica circulaţiei (supravegherea spaţiului aerian, conducerea unor autovehicule autonome).

Produsele acestor cercetări nu mai sînt programe uriaşe rulînd pe supercalculatoare de tip Cray sau Suprenum, ci devin programe scrise pentru hard-uri special concepute pentru a efectua funcţii inteligente. Aceste hard-uri speciale vor fi compuse din mai multe mii de procesoare legate între ele ca neuronii, şi care se vor constitui într-o reţea neuronală (ca noduri şi sinapse).

Una dintre funcţiile inteligente care se doreşte a fi realizată este aceea de a se putea descrie orice problemă matematică nu prin program ci prin autoorganizarea reţelelor neuronale. Acest proces de autoorganizare ar trebui să aibă loc pe parcursul unui proces de învăţare obţinut prin cooperarea unei topologii iniţiale, a unor reguli de învăţare şi a unui număr mare de antrenamente. Acest proces de învăţare este, fără îndoială, principalul scop al cercetării în Neuroinformatică.

Alături de aceste reţele autoorganizabile pomenite pînă acum, reţele sprijinite de lucrările unor specialişti ca T. Kahonen, S. Grossberg, Malsburs şi alţii, a apărut şi o altă clasă - aşa numitele reţele supervizoare.

Chiar dacă existenţa lor nu este motivată biologic aşa cum sînt celelalte prin neuroni, sinapse, faze de învăţare, reţelele supervizoare sînt totuşi foarte asemănătoare cu primele. Aceste reţele se bazează în principal pe ideea de a face cîteva încercări care să poată fi generalizate după aceea.

O probă de antrenament constă dintr-o mostră oarecare (de exemplu o combinaţie de pixeli ai literei "A") şi din scopul dorit al identificării (în exemplul nostru, obţinerea codului ASCII corespunzător literei "A"). Generalizarea ar consta în aceea că nu toate combinaţiile de pixeli posibile pentru litera "A" trebuiesc memorate. De fapt va fi nevoie doar de cîteva (bine înţeles foarte diferite şi bine alese) probe de antrenament, cu ajutorul cărora reţeaua să înveţe o 'imagine universală' a literei "A". Pe parcursul expunerii vom mai oferi şi alte exemple.

Rădăcinile comune ale celor două modele de reţele se găsesc mult înapoi în timp; în anul 1943 McCulloch şi Pitts au dezvoltat un formalism matematic pentru descrierea reţelelor neuronale, şi au demonstrat că pentru orice problemă de rezolvat se poate dezvolta o reţea neuronală care să o rezolve. Lucrările lor au constituit bazele pentru teoria automatelor finite, domeniu atît de important astăzi. Din păcate nu s-a dat nici o indicaţie asupra metodelor de a rezolva problema în mod automat.

Un prim algoritm de învăţare (motivat biologic) a fost dat în 1949 de Dr. Hebb: puterea legăturii dintre doi neuroni creşte atunci cînd amîndoi sînt activi în acelaşi timp. Această formulare nu poate fi înţeleasă izolat - nu precizează ce înseamnă noţiunile de neuron, putere de legătură, activitate şi simultaneitate - dar dă totuşi un nucleu care s-a păstrat într-o formă uşor modificată pînă în abordările actuale ale teoriei reţelelor neuronale.

În anii '60 Widrow şi Hoff au dezvoltat modelul neuronal adaptiv

Adaline (numit mai tîrziu Madaline de la "multiple-Adaline") în timp ce F. Rosenblatt dezvolta modelul "Percepton". Cei doi au demonstrat noile proprietăţi ale reţelelor: ele se puteau autoorganiza parţial, erau capabile de autoînvăţare, puteau clasifica mostre simple şi chiar puteau generaliza experienţe la modul prezentat mai sus. Se puteau deci recunoaşte probe pe care reţeaua nu fusese antrenată anterior.

Principalele slăbiciuni ale acestui tip de reţele de supervizare au fost evidenţiate în 1969 de către M. Minski şi S. Papert. Abia în anii '80 au fost redescoperite aceste reţele şi scoase la lumină de către Rumelhardt şi Hinton care au organizat reţelele în mai multe straturi şi au pus la punct un algoritm de învăţare ingenios numit algoritmul de propagare înapoi.

Acest algoritm a adus o schimbare majoră, cu ajutorul lui reţelele reuşind să depăşească toate obiecţiile lui Minski. Totuşi timpii de convergenţă ai algoritmilor de învăţare (deci timpul de învăţare) rămîn destul de mari.

Cum funcţionează aceşti algoritmi de propagare înapoi vom exemplifica mai tîrziu prin numeroase exemple.

Această primă parte a cursului a încercat o succintă introducere la dezvoltarea şi diversele stadii prin care a trecut cercetarea în acest domeniu. articolul al doilea se va ocupa de algoritmul de propagare înapoi şi de topologia de reţea aferentă. Vom realiza aceasta prin exemple şi nu printr-o prezentare teoretică aridă.

În partea a treia ne vom referi la o aplicaţie anume a reţelelor neuronale: sistemele expert. Spre deosebire de abordările clasice ale sistemelor expert (de tip Prolog), de data aceasta ne vom referi la regulile de producţie de tipul "dacă A este adevărat atunci rezultă B". Vom vedea că noua abordare este mult mai flexibilă şi mult mai tolerantă la greşeli. Această particularitate a reţelelor neuronale, de a asocia la seturi vagi de date de intrare, date de ieşire ferme, o vom folosi la stocarea informaţiilor într-o bază de date.

În ultima parte a seriei vom vorbi despre necesitatea combinării reţelelor neuronale cu alte tehnici mai moderne. De cele mai multe ori problemele practice sînt greu de rezolvat folosind în exclusivitate reţele neuronale. Problemele trebuie descompuse, în general, în probleme mai simple pe care să le prelucrăm separat cu ajutorul reţelelor neuronale sau cu alte tehnici.

Un exemplu tipic pentru aceasta este recunoaşterea scrisului, transpunerea unei coli de hîrtie plină de semne în codurile ASCII corespunzătoare pe care le poate înţelege calculatorul. Pentru început contrastul alb-negru de pe hîrtie este digitizat cu ajutorul unui scanner. În imaginea rezultată trebuie diferenţiat mai întîi textul de restul graficii, apoi în interiorul textului trebuie delimitată fiecare literă. Abia după aceea se poate începe identificarea propriu-zisă a semnelor cu codurile ASCII corespunzătoare.

Dar despre toate acestea în numerele viitoare ale revistei.

(PC+Technik 2/91, Peter Schmitz)

Formate grafice

(continuare din pagina 15)

EPS, PS

PS, respectiv EPS, reprezintă prescurtarea pentru "Postscript" sau "Encapsulated Postscript". Acesta este un format foarte important şi foarte răspîndit în domeniul Desktop Publishing. Formatul EPS nu este nici el formatul de bază al unui anumit program, ci este un format în care pot fi create fişiere din diferite programe de grafică, cum ar fi: "Arts & Letters", "Corel Draw", "Adobe Illustrator" (pe Mac, dar în curînd şi pe PC), şi de "Aldus Freehand" (pe Mac). Aceste fişiere conţin informaţii despre grafice sau texte în formatul limbajului de descriere a paginilor "Postscript". Aceste informaţii sînt folosite pentru imprimarea pe imprimante laser sau pe exponatoare Linotronic.

Deoarece Postscript este un limbaj puternic de descriere, care cunoaşte practic toate elementele de grafică şi de text şi toate modalităţile de prezentare, el poate importa aceste fişiere din toate programele importante DTP. Astfel obiectele grafice importate apar în dimensiunea lor reală şi în poziţia dorită şi pot fi tipărite pe imprimante compatibile Postscript foarte corect şi cu rezoluţia maximă.

În programele DTP - cu excepţia măririlor/micşorărilor - nu se mai pot face alte manipulări (de aici şi denumirea "Encapsulated"). Pentru a putea vedea pe ecran, din interiorul unui program DTP, conţinutul unui astfel de fişier, fişierele EPS pot conţine pe lîngă informaţiile referitoare la vectori şi informaţii suplimentare, cum ar fi o imagine rudimentară a unei grafici bitmap. Programele DTP, cum ar fi Pagemaker sau Ventura Publisher, pot vizualiza pe ecran o astfel de imagine. Dacă aşa-zisul header (antet) TIFF lipseşte, pe ecran va apare doar o casetă gri.

HPGL

HPGL este prescurtarea pentru "Hewlett-Packard Graphics Language" şi este un format asemănător cu Postscript, care conţine instrucţiuni de comandă pentru o imprimantă sau un plotter, cu deosebirea că nu se foloseşte limbajul de descriere Postscript, ci un limbaj de descriere total diferit, limbajul special al imprimantelor şi plotter-elor fabricate de firma Hewlett-Packard.

(Computer Live 2/91, Klaus Schrdl)

Obiectele se încapsulează

Dacă doriţi să programaţi mai comod şi să lucraţi mai uşor cu proiecte mari, atunci nu puteţi face abstracţie de OOP (programarea orientată obiect).

Pînă acum în programare se făcea o distincţie netă între structurile de date şi funcţiile care lucrau cu ele. Tehnicile de programare bazate pe acest principiu au fost verificate în practică. Odată cu sporirea complexităţii problemelor care se cereau rezolvate, s-a văzut însă că aceste metode trebuie modificate, dezvoltate în continuare, sau trebuie găsite metode cu totul noi. Ideea de a combina datele şi metodele lor de prelucrare nu este nouă, dar abia odată cu introducerea unui limbaj de programare orientat obiect (C++), pe un microcalculator, a dat o semnificaţie sporită tehnicii OOP. Filozofia programării orientate obiect se abate de la metodele obşnuite, în plus ea introduce o serie întreagă de termeni noi care ar putea îngreuna înţelegerea.

Merită deci să ne ocupăm mai îndeaproape de acest domeniu. Însuşirile obiectelor facilitează o elaborare economică a programelor şi uşurează, (mai ales), întreţinerea bibliotecilor de programe. Utilizînd cu consecvenţă tehnicile OOP, multe programe devin mai transparente şi pot fi cuprinse mai uşor dintr-o privire.

Un tip obiect este un nou tip de date. Definiţia sa este dată de utilizator, deci un obiect este un "tip de date definit de utilizator". Odată definit un obiect, cu el se poate opera la fel ca şi cu toate celelalte tipuri de date recunoscute de limbajul de programare în care se lucrează. Declararea unui obiect se face ca şi declararea unei variabile normale. Compilatorul îi rezervă o zonă de memorie care poate fi ulterior accesată prin intermediul numelui obiectului.

Cum arată însă un obiect concret? În Pascal el este definit şi declarat în mod asemănător cu un articol (record). La început trebuie stabilit deci un tip obiect la care să se poată raporta ulterior un obiect.

Listingul OOP_prima_incercare vă arată cum funcţionează aşa ceva în Turbo-Pascal. În partea de definire a tipului este definit obiectul cu numele Caracter. Deosebirea vizibilă faţă de o declaraţie de articol o constituie utilizarea cuvîntului rezervat OBJECT. Toate variabilele sînt cuprinse sub un identificator, ca şi în cazul unui articol. Un obiect este însă mai mult: sub acelaşi acoperiş sînt unite nu numai datele ci şi metodele lor de prelucrare. Tipul de obiect Caracter permite patru metode de prelucrare: "stabileste_pozitie", "citeste_pozitie", "scrie" şi "citeste". Toate aceste metode influenţează sau utilizează variabilele interne "linia" şi "coloana". La stabilirea unui tip obiect se spune doar ce face o metodă, este numită şi i se specifică parametrii. Metodele de prelucrare propriu-zise sînt declarate înafara declaraţiei tip obiect - în listingul OOP_prima_incercare se poate vedea şi cum. Acest lucru se întîmplă în acelaşi mod ca şi în cazul declaraţiilor obişnuite de funcţii şi proceduri. Numele este format din numele tipului obiect şi din numele metodei, separate printr-un punct. Puteţi lucra deci aşa cum sînteţi obişnuiţi de la lucrul cu articole. Puteţi utiliza chiar şi ins-

Planificarea cursurilor

Cursul 1: Noţiuni şi termeni; definţii şi însuşiri ale obiectelor; încapsulare şi moştenire

Cursul 2: Declaraţii dinamice; polimorfism

Cursul 3: Constructor şi Destructor; metode virtuale

Cursul 4: Ierarhii de obiecte; diferite tehnici OOP

Cursul 5: Exemple de OOP.

Tip obiect: Un tip obiect este un tip de date definit de utilizator. În el sînt definite structurile de date şi metodele de prelucrare corespunzătoare.

Obiect: Un obiect este stabilit şi manipulat ca şi o variabilă. Compilatorul rezervă deci pentru un obiect spaţiul de memorie necesar, care este accesibil prin intermediul numelui obiectului.

Metodă: Pentru a putea lucra cu datele stabilite în interiorul unui obiect, în cursul stabilirii lui trebuie declarate şi funcţiile şi procedurile corespunzătoare. Tocmai acestea însă sînt metodele.

Sarcină: După stabilirea unui obiect, numele unei metode este utilizat pentru a-i transmite o sarcină acestuia.

Protocol: Totalitatea sarcinilor accesibile unui tip obiect este desemnată ca protocol.

Încapsulare: Cuprinderea împreună a structurilor de date şi a metodelor de prelucrare se numeşte încapsulare. Definirea obiectelor în cadrul unui Unit întăreşte încapsularea.

Moştenire: Tipurile obiect îşi pot transmite toate însuşirile unor noi obiecte. Această modalitate de moştenire elimină necesitatea de a redefini metodele deja definite. Trebuie stabilite doar ceea ce s-a schimbat faţă de obiectul "mamă" sau care sînt datele şi metodele care trebuie adăugate.

Ierarhie de obiecte: Deoarece tipurile obiect îşi pot moşteni însuşirile, se poate construi o ierarhie arborescentă a tuturor urmaşilor. Toate tipurile obiect ale acestei ierarhii sînt legate prin "legături de rudenie". De datele şi metodele obiectului rădăcină pot dispune toţi urmaşii.

```
program OOP_prima_incercare;

const Seg_BWS = $B800;
{Adresa segmentului pentru prezenta-
rea culorilor}

{Stabilirea unui obiect}
type
  Caracter = OBJECT
    linia,coloana : byte;
    procedure stabileste_pozitie
                  (x,y : byte);
    procedure citeste_pozitie
                  (var x,y : byte);
    procedure citeste (var ch : char);
    procedure scrie (ch : char);
  end;

{Stabilirea metodelor pentru obiectul
Caracter}

procedure Caracter.stabileste_pozitie
                       (x,y : byte);
begin
  coloana := x;
  linia := y;
end;

procedure Caracter.citeste_pozitie
                  (var x,y : byte);
begin
  x := coloana;
  y := linia;
end;

procedure Caracter.citeste
                  (var ch : char);
var Ofs_BWS : word;
begin
  if ((linia>25) or (coloana>80)) then
exit;
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  ch := chr (mem[Seg_BWS:Ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var Ofs_BWS : word;
begin
  if ((linia>25) or (coloana>80))
                       then exit;
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

{Declaratia unui obiect}
var
  z : Caracter;

{Programul principal}
begin
  z.stabileste_pozitie (10,10);
  z.scrie ('A');
end.

{LISTING1.PAS  Sfirsit}
```

```
type

Sir_de_caractere = OBJECT (Caracter)
  procedure citeste_text
              (var st : string;
          lng : byte);
  procedure scrie_text (st : string);
end;

{Listing 2}

Prin mostenire obiectul
Sir_de_caractere pastreaza toate insu-
sirile obiectului Caracter.
O redefinire nu este necesara.
Listingul 3 are deci aceeasi semnifi-
catie ca si listingul 2.
```

```
type

Sir_de_caractere = OBJECT
  linia, coloana : byte;
  procedure stabileste_pozitie
                  (x,y : byte);
  procedure citeste_pozitie
                  (var x,y : byte);
  procedure scrie (ch : char);
  procedure citeste (var ch : char);
  procedure scrie_text (st : string);
  procedure citeste_text
                  (var st : string;
                   lng : byte);
end;

{LISTIN23.PAS Sfirsit}
```

trucţiunea "with" pentru a avea mai puţin de scris.

Surprinzător este faptul că variabilele interne "linia" şi "coloana" pot fi modificate numai prin intermediul metodelor "stabileste_pozitie" şi "citeste_pozitie". Utilizarea acestor variabile înafara metodelor obiectului nu este posibilă. Ceea ce la prima vedere pare incomod, căci s-ar fi

putut şi mai direct, constituie de fapt o însuşire de bază a programării orientate obiect. Înmănuncherea structurilor interne de date cu metodele corespunzătoare de prelucrare se cheamă încapsulare. Această noţiune sugerează faptul că pentru utilizator structura obiectului tip este neesenţială şi inaccesibilă. Pentru utilizator obiectul tip există doar sub forma tuturor metodelor de acces la el. Cum funcţionează propriu-zis o metodă nu este interesant, fiind suficient să ştim ce se poate face cu metodele. În acest mod structura internă a datelor poate fi protejată împotriva unor accese nedorite.

```
program OOP_a_doua_incercare;

const Seg_BWS = $B800;
{Adresa segment monitor color}

{Doua tipuri de obiecte}
type
  Caracter = OBJECT
    linia,coloana : byte;
    procedure stabileste_pozitie
                  (x,y : byte);
    procedure citeste_pozitie
                  (var x,y : byte);
    procedure citeste (var ch : char);
    procedure scrie (ch : char);
  end;

Sir_de_caractere = OBJECT (Caracter)
  procedure citeste_text
                  (var st : string;
                    lng : byte);
  procedure scrie_text (st : string);
end;

{Metodele tipului de obiecte Caracter}

procedure Caracter.stabileste_pozitie
                  (x,y : byte);
begin
  coloana := y;
  linia := x;
end;

procedure Caracter.citeste_pozitie
                (var x,y : byte);
begin
  x := coloana;
  y := linia;
end;

procedure Caracter.citeste
                (var ch : char);
var Ofs_BWS : word;
begin
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  ch := chr (mem[Seg_BWS:Ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var Ofs_BWS : word;
begin
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

{Metodele tipului de obiecte
Sir_de_caractere}

procedure Sir_de_caractere.scrie_text
                  (st : string);
var i,j : integer;
begin
  j := coloana;
  for i := 1 to length (st) do begin
    scrie (st[i]);
    inc (coloana);
  end;
  coloana := j;
end;

procedure
    Sir_de_caractere.citeste_text
  (var st : string; lng : byte);
  var  i : integer;
       ch: char;
begin
  st := '';
  for i := 1 to lng do begin
    citeste (ch);
    st := st + ch;
    inc (coloana);
  end;
  coloana := coloana - lng;
end;

{Declararea obiectelor si
        variabilelor necesare}
var  z : Caracter;
  t : Sir_de_caractere;
  st: string;

{Programul principal}
begin
  z.stabileste_pozitie (10,10);
  z.scrie ('A');
  t.stabileste_pozitie (12,50);
  t.scrie_text ('Mostenim insusiri');
  t.citeste_text (st,12);
  t.stabileste_pozitie (14,50);
  t.scrie_text (st);
end.

{LISTING4.PAS Sfirsit}
```

```
unit t_handle;

InterFace

{Adresa segment monitor color}
const Seg_BWS = $B800;

{Doua tipuri de obiecte}
type
  Caracter = OBJECT
    linia,coloana : byte;
    procedure stabileste_pozitie
                    (x,y : byte);
    procedure citeste_pozitie
                    (var x,y : byte);
    procedure citeste (var ch : char);
    procedure scrie (ch : char);
  end;

Sir_de_caractere = OBJECT (Caracter)
  procedure citeste_text
                  (var st : string;
                       lng : byte);
  procedure scrie_text (st : string);
end;

Implementation

{Metodele tipului de obiecte Caracter}

procedure Caracter.stabileste_pozitie
                    (x,y : byte);
begin
  coloana := y;
  linia := x;
end;

procedure Caracter.citeste_pozitie
                  (var x,y : byte);
begin
  x := coloana;
  y := linia;
end;

procedure Caracter.citeste
                    (var ch : char);
var Ofs_BWS : word;
begin
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  ch := chr (mem[Seg_BWS:Ofs_BWS]);
end;

procedure Caracter.scrie (ch : char);
var Ofs_BWS : word;
begin
  Ofs_BWS:=2*(coloana-1)+160*(linia-1);
  mem[Seg_BWS:Ofs_BWS] := ord (ch);
end;

{Metodele tipului de obiecte
Sir_de_caractere}

procedure Sir_de_caractere.scrie_text
                        (st : string);
var i,j : integer;
begin
  j := coloana;
  for i := 1 to length (st) do begin
    scrie (st[i]);
    inc (coloana);
  end;
  coloana := j;
end;

procedure
      Sir_de_caractere.citeste_text
  (var st : string; lng : byte);
var  i : integer;
  ch: char;
begin
  st := '';
  for i := 1 to lng do begin
    citeste (ch);
    st := st + ch;
    inc (coloana);
  end;
  coloana := coloana - lng;
end;

end.

{LISTING5.PAS Sfirsit}
```

Unui obiect i ne adresăm dîndu-i o sarcină, tot restul lăsînd în seama lui. Listingul nostru OOP_prima_incercare dă lămuriri şi în această privinţă. Obiectului Z i-am dat sarcina "stabileste_pozitie(10,10)", realizarea lăsînd-o în seama lui. Următoarea instrucţiune transmite sarcina "scrie('A')". Numele metodei împreună cu eventualii parametrii nu sînt deci altceva decît sarcini pe care obiectul le poate înţelege. Totalitatea sarcinilor care pot fi executate de un obiect se numeşte şi protocol.

Dar programarea orientată obiect nu are doar însuşirea încapsulării. Ar fi şi prea puţin.

Dacă, de ex., doriţi să utilizaţi obiectul Caracter pentru a opera cu un şir, mînuirea lui devine incomodă. Dacă însă nu doriţi să renunţaţi la însuşirile obiectului Caracter, pentru că de ex. funcţionează excelent şi fără greşeală, atunci le puteţi lăsa moştenire unui alt obiect, căruia puteţi să-i mai adăugaţi şi alte însuşiri. Din Listingul 2 se poate vedea cum se moşteneşte. Noul tip obiect "Sir_de_caractere" dispune de toate metodele "mamei" sale. De unde se trage obiectul Sir_de_caractere se poate vedea din instrucţiunea de definiţie:

```
program OOP_a_treia_incercare;

uses t_handle;

var  z : Caracter;
   t : Sir_de_caractere;
   st: string;

begin
   z.stabileste_pozitie (10,10);
   z.scrie ('A');
   t.stabileste_pozitie (12,50);
   t.scrie_text ('Mostenim insusiri');
   t.citeste_text (st,12);
   t.stabileste_pozitie (14,50);
   t.scrie_text (st);
end.
{LISTING6.PAS Sfirsit}
```

```
type
   <Nume> = OBJECT
      <Lista datelor>
      <Lista metodelor de prelucrare>
   end;

procedure <Metoda>
            [(<Lista de parametrii>)]

procedure <Tip obiect>.<Metoda>
               (<Lista de parametrii>)

type
   <Nume> = OBJECT (<Nume obiect mama>)

{OOP112.PAS Sfirsit}
```

Sir_de_caractere =
OBJECT (Caracter)

Totodată se mai definesc două noi însuşiri, şi anume "citeste_text" şi "scrie_text".

Moştenirea are cîteva avantaje. O nouă definire a metodelor moştenite nu mai este necesară. Obiectul Sir_de_caractere din Listingul 2 este ca urmare identic cu cel din Listingul 3. Metodele deja moştenite pot fi modificate şi dezvoltate în continuare, totuşi aceasta este deja o anticipare. În acest loc ne mulţumim să vă prezentăm un exemplu care vă va arăta cum funcţionează mecanismul de moştenire.

Să facem o a doua încercare cu Listingul 4. La început, în partea de definire, se definesc două tipuri obiect. Tipul obiect Caracter este deja cunoscut din Listingul 1. Sir_de_caractere provine din tipul Caracter, în mod clar vizibil după "mama" cuprinsă între paranteze. La sfîrşit urmează stabilirea tuturor metodelor. Programul principal demonstrează în mod foarte clar faptul că obiectul Sir_de_caractere a mostenit însuşirile obiectului Caracter. Cu toate că nu a fost stabilită explicit, metoda "stabileşte_pozitie" poate fi transmisă şi obiectului Sir_de_caractere.

Pînă în acest moment v-am prezentat regulile jocului în cazul programării orientate obiect, în orice caz partea care ne-am propus-o.

Tipurile obiect sînt stabilite ca şi articolele în partea de definire, dar aici sînt unite sub acelaşi acoperiş atît datele cît şi metodele lor de prelucrare. Această caracteristică se numeşte încapsulare. (Fig. OOP112.PAS, liniile 1-5).

Lista metodelor de prelucrare descrie doar, ceea ce fac metodele individuale, deci numele şi eventualii parametrii (Fig. OOP112.PAS, linia 7).

Numele unui obiect este de fapt sarcina cu care poate fi activat obiectul referit. Metodele propriu-zise sînt stabilite înafara tipului obiect, totuşi sînt legate de acesta. Numele tipului obiect precede definirea numelui metodei. (Fig. OOP112.PAS, linia 9).

Tipurile obiect pot lăsa moştenire anumite însuşiri unor obiecte noi. Noului tip obiect trebuie să i se specifice provenienţa
(Fig. OOP112.PAS, linia 11-12).

În noul obiect tip pot fi stabilite noi date şi metode, acesta posedînd astfel atît însuşirile vechi cît şi pe cele noi.

Să revenim încă odată asupra faptului că obiectele, odată stabilite, pot fi utilizate la fel cu celelalte variabile. Desigur aceste stabiliri pot fi făcute şi în interiorul unei unităţi (Units). Să mai încercăm odată programul cunoscut, folosind de această dată Listingul 5. Nu trebuie decît să declarăm sursa ca şi Unit şi să stabilim segmentele Interface şi Implementation. Se poate observa că în partea de Interface nu sînt prezente decît definiţiile celor două obiecte tip. Stabilirea tuturor metodelor urmează în partea Implementation. Rezultatul unei astfel de construcţii este o întărire a încapsulării, deoarece în acest caz şi metodele vor fi ascunse. Pe această cale utilizatorul obiectelor află doar ce poate face cu metodele, modul lor de funcţionare rămînînd necunoscut.

Programul din Listingul 6 este identic cu cel din Listingul 4. Definiţiile obiectelor au fost ascunse într-un Unit în Listingul 5.

Pînă acum am stabilit obiectele doar static. Bineînţeles, în Turbo-Pascal, obiectele potr fi manipulate şi dinamic. Sintaxa este aceeaşi atît în cazul declaraţiilor statice cît şi a celor dinamice, trebuie doar avută grijă ca să se respecte regulile jocului în programarea dinamică, cum ar fi rezervarea de memorie, eliberarea memoriei, să nu se piardă din vedere simbolurile speciale ale limbajului.

Dar despre acest lucru vom vorbi, printre altele, în episodul următor. Acolo vă vom prezenta modul în care obiectele pot fi tratate dinamic.

(PC+Technik 2/91,
Andreas Schimpf)

Sisteme de achiziţie de date şi control cu calculatoare personale compatibile IBM PC/XT/AT

Majoritatea dintre noi, am fi de acord cu faptul că PC-urile au avut o mare influenţă asupra tehnicilor de măsurare, control, analiză şi prezentare. Din păcate însă, PC-ul nu poate fi considerat un "RĂSPUNS" la toate întrebările noastre, ci mai degrabă o "SCULĂ". În ideea de a folosi cît mai eficient această sculă (tehnologie), trebuie mai întîi să-i cunoaştem caracteristicile, capabilităţile, puterea, dar şi LIMITELE sale.

Poate una dintre cele mai importante limitări se referă la TIPUL SEMNALELOR pe care PC-ul le poate accepta şi prelucra. Aceasta, deoarece majoritatea semnalelor din lumea reală NU pot fi conectate direct la PC. Echivalenţii electrici ai temperaturii, presiunii, vitezei, debitului şi a stărilor ÎNCHIS/DESCHIS, nu sînt direct compatibile cu cerinţele PC-urilor. Cum intră şi ies atunci semnalele în şi din PC?

Acest articol prezintă informaţiile şi recomandările de bază necesare privind achiziţia şi controlul datelor, cu calculatoare personale compatibile PC/XT/AT.

Achiziţii de date şi control: o privire de ansamblu

Termenul de ACHIZIŢIE DE DATE ŞI CONTROL - AD&C (data acquisition and control - DA&C), poate fi înţeles diferit de fiecare dintre noi. Supravegherea unui voltmetru şi înregistrarea manuală a valorii citite, reprezintă o ACHIZIŢIE DE DATE. Pe de altă parte, dacă aprindem lumina într-o încăpere, înseamnă că am efectuat o operaţie de CONTROL.

În concluzie, achiziţia de date, este procesul de colectare a informaţiilor care descriu o anumită situaţie. Datele culese reflectă ce s-a întîmplat în momentul îndeplinirii unei anumite condiţii, care poate fi definită cu ajutorul unui EVENIMENT.

Sistemele în timp real se caracterizează prin abilitatea lor de a executa un anumit TASK de AD&C într-o anumită FEREASTRĂ DE TIMP. TIMPUL DE RĂSPUNS al sistemului depinde de viteza şi precizia cerută de aplicaţia respectivă. Pentru orice aplicaţie de AD&C, oricît de lentă ar fi, există o aplicaţie suficient de lentă, care pentru acea aplicaţie reprezintă un sistem în timp real.

CONTROLUL implică generarea unui semnal de ieşire ca răspuns la un semnal de intrare. Controlul poate fi de tip : BUCLĂ - DESCHISĂ (OPEN - LOOP) sau BUCLĂ ÎNCHISĂ (CLOSED LOOP).

Oprirea încălzirii la ora 4:00 p.m. este un exemplu de control în buclă deschisă, în timp ce oprirea încălzirii din cauză că este prea cald este un control în buclă închisă.

În general, datele se culeg de ingineri, tehnicieni, fizicieni, chimişti sau alţi oameni implicaţi în cercetare, test, dezvoltare, producţie, calitate, control, management etc., pentru a produce analize, rapoarte, a lua decizii şi a executa operaţii de control. Datele sînt înregistrate, analizate, utilizate şi comunicate. Metodele de lucru, însă, se pot schimba. Acum ne concentrăm asupra faptului de a folosi maşinile pentru a rezolva multe din problemele noastre de achiziţie de date şi control.

Motivul principal este CREŞTEREA PRODUCTIVITĂŢII. Viteza, precizia, fiabilitatea şi costul sînt doar factori colaterali. Astfel, sistemele de astăzi se bazează pe cea mai productivă maşină - calculatorul. Figura 1 ne prezintă componentele unui sistem de AD&C. Calculatorul realizează nu numai analiza şi luarea deciziilor ci şi controlul funcţiilor active de preprocesare şi conversie a semnalelor.

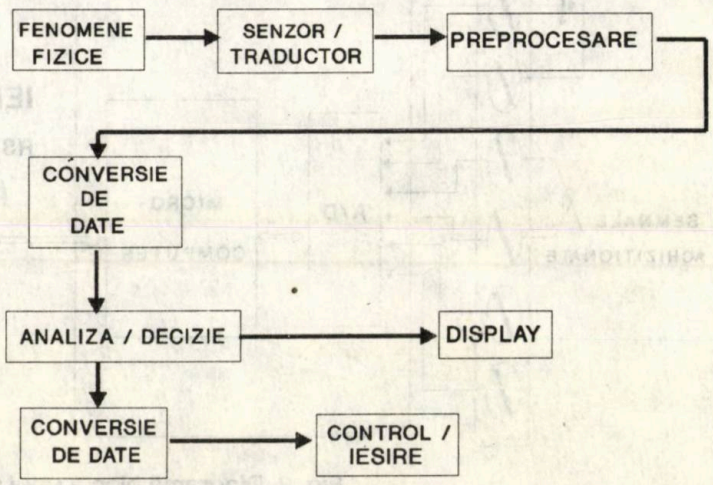


Fig. 1. Schema unui sistem de achiziţie date şi control

Calculatoarele moderne oferă viteză ridicată, flexibilitate, adaptabilitate, fiabilitate şi cantitate mare de memorie, proprietăţi ce permit realizarea unor funcţii ca: analiză, stocare, afişare, generare de rapoarte, control şi comunicaţie. Oricum, însă, cele mai multe semnale din lumea înconjurătoare (temperatură, presiune, debit, viteză, intensitate, poziţie etc.) nu pot fi citite direct de calculatoarele numerice. Aceşti parametri sînt reprezentaţi de semnale analogice care se caracterizează prin nivelul lor CONTINUU.

Calculatorul poate recunoaşte numai nivele DIGITALE (închis sau deschis). De aceea este nevoie de dispozitive ce realizează o conversie a acestor semnale.

LEGĂTURA cu lumea reală - Produsele de AD&C convertesc semnalele din lumea reală într-un format acceptat de calculatoarele numerice. Sistemele de AD&C pot regenera semnale analogice precum şi alte semnale pe baza instrucţiunilor de program realizînd astfel, puntea de legătură între calculator şi lumea reală.

PC-urile, în comparaţie cu alte tipuri de calculatoare, s-au dezvoltat cel mai rapid spre a deveni celula de bază a sistemelor de AD&C.Ele au cucerit deja diferite domenii în acest sens cum ar fi: culegerea de date şi automatizări în laborator, culegerea de date de la aparate medicale şi monitorizarea pacienţilor, testarea automată a echipamentelor (Automatic Test Equipment - ATE, care include inspecţia, testul de anduranţă, testul de producţie şi testarea finală), monitorizarea şi controlul proceselor industriale.

Diferite tipuri de sisteme şi modurile de conectare a lor la un PC

O parte importantă a oricărui sistem de AD&C este calculatorul GAZDĂ (HOST computer). Există două posibilităţi de interfaţare HARD a sistemelor de AD&C cu calculatorul: conectare directă pe magistrala PC-ului (produse pentru BUS-ul INTERN) şi conectarea prin intermediul unui canal de comunicaţie standard ca de exemplu RS-232, RS-422 sau IEEE-488 (produse pentru BUS-ul EXTERN). Fiecare dintre cele două are avantaje şi dezavantaje.

PRODUSE PENTRU BUS EXTERN - Există numeroase avantaje asociate acestor produse cum ar fi: VIRTUAL, putem configura orice sistem, oricît de mare dorim; sistemul de AD&C poate fi plasat la distanţă de calculatorul gazdă (şi astfel cît mai aproape de traductori); sistemul de AD&C poate degreva calculatorul gazdă de unele TASK-uri de culegere-date; sistemul de AD&C poate fi interfaţat virtual cu orice tip de calculator.

Figura 2 ne arată o diagramă bloc simplificată a unui sistem de AD&C bazat pe BUS extern. Comunicaţia prin RS-232,RS-422 sau IEEE-488, necesită ca sistemul de AD&C să aibă microprocesorul lui propriu. Acest microcalculator local, facilitează de asemenea şi operaţiile de comunicare la distanţă, reducînd astfel sarcinile PC-ului gazdă.

Sistemele de acest tip sînt concepute ca sisteme independente, avînd incintele lor proprii în care sînt montate, şi care conţin nu numai microsistemul ci şi sursa de alimentare proprie precum şi hardul necesar pentru intrările şi ieşirile analogice şi numerice. De obicei, funcţiile de intrare/ieşire (Input/Output I/O) sînt grupate pe plăci separate, ceea ce permite o configurare mai rapidă a sistemelor. Sistemele de AD&C cu BUS extern, permit construcţia sisteme-

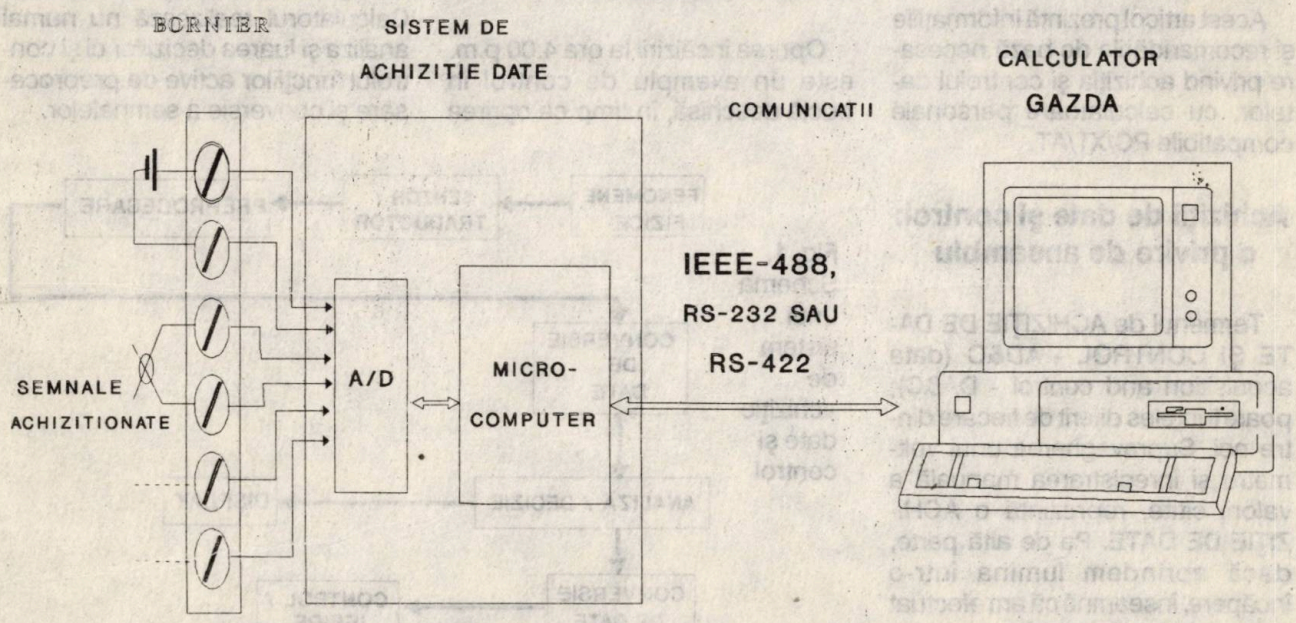


Fig. 2. Diagramă bloc a unui sistem AD&C cu bus extern

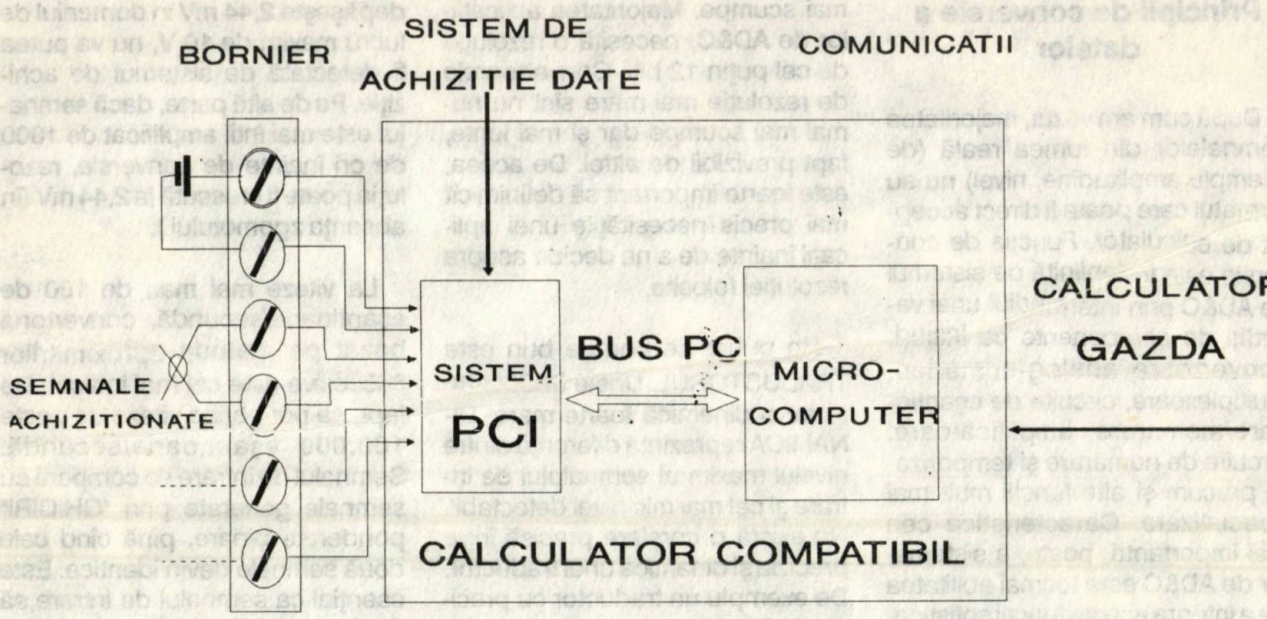


Fig. 3. Sistem AD&C cu bus intern

lor distribuite care pot realiza monitorizarea şi controlul unui mare număr de parametri aflaţi la distanţă mare unul de celălalt şi la mare distanţă de PC-ul gazdă.

PRODUSE PENTRU BUS INTERN - Principalele avantaje ale conectării direct pe BUS-ul intern al PC-ului includ: viteză mare, cost scăzut, dimensiuni mai mici. Aceste tipuri de sisteme de AD&C reduc costul pentru că nu mai au nevoie de incinte şi surse de alimentare separate, alimentarea făcîndu-se din sursa internă a PC-ului. Vitezele mari se obţin datorită eliminării protocolului de comunicaţie, relativ lent, folosit. De exemplu, rata de achiziţie a unui sistem de AD&C, folosind interfaţa de comunicaţie RS-232 la viteza de 9600 Baud, este limitată la aproximativ 20 de citiri analogice/secundă. Pe de altă parte, unele produse de AD&C pentru BUS intern, pot achiziţiona şi la rate de 1 milion de eşantioane/secundă. Figura 3 ne prezintă o diagramă bloc simplificată a unui sistem de AD&C bazat pe BUS intern.

Toate sistemele de AD&C pentru BUS intern sînt compuse din plăci de interfaţă ce se conectează direct pe SLOT-urile din PC. Unele sînt proiectate cu un număr FIX de intrări/ieşiri, analogice/numerice, ceea ce prezintă dezavantajul că ele nu permit adăugarea ulterioară a unor funcţii noi sau a unor canale noi de monitorizat.

În contrast, sistemele MODULARE permit utilizatorului să selecteze numărul de funcţii de I/E dorite. Această posibilitate este concepută ca o familie de module de funcţii care prezintă o flexibilitate mai mare, dînd posibilitatea definirii oricărei aplicaţii specifice.

PC-ul şi sistemele de achiziţii de date şi control

Din punct de vedere istoric, industrial şi ştiinţific, aplicaţiile de tip AD&C au fost implementate mai întîi, cu ajutorul calculatoarelor mari, (MAINFRAME). sau cu ajutorul minicalculatoarelor. Acestea erau maşini de 16 biţi care dispuneau de sisteme de operare multitasking şi multiuser Complexitatea şi costul acestora dicta ca aceste sisteme să fie configurate ca sisteme centrale, cu mai mulţi utilizatori. Lucrările mici nu justificau însă această investiţie mare.

Apariţia calculatoarelor personale - PC, a făcut teoretic posibil, ca oricine să beneficieze de flexibilitate, putere de calcul şi eficienţa sistemelor de AD&C. PC-urile oferă performanţe ridicate şi cost scăzut pe lîngă o uşurinţă în utilizare fără precedent. Datorită unui nivel ridicat de standardizare între constructorii de PC-uri şi cei de sisteme de AD&C există astăzi posibilitatea ca un om de ştiinţă sau inginer să-şi poată implementa sistemul său propriu de AD&C într-un timp foarte scurt. Figura 4 ne arată relaţia dintre sistemul de AD&C şi calculatorul gazdă.

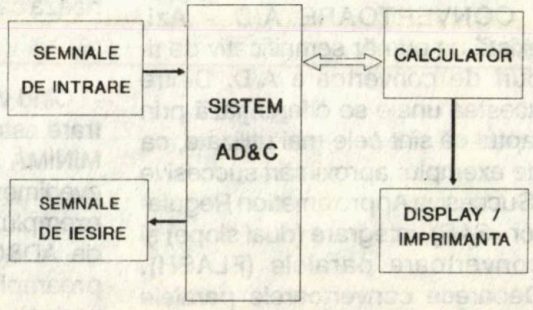


Fig. 4. Un sistem AD&C computerizat

Principii de conversie a datelor

După cum am văzut, majoritatea semnalelor din lumea reală (de exemplu amplitudine, nivel) nu au formatul care poate fi direct acceptat de calculator. Funcţia de conversie este îndeplinită de sistemul de AD&C prin intermediul unei varietăţi de componente ce includ: convertoare analog-numerice, multiplexoare, circuite de eşantionare/memorare, amplificatoare, circuite de numărare şi temporizare precum şi alte funcţii mult mai specializate. Caracteristica cea mai importantă, poate, a sistemelor de AD&C este tocmai abilitatea de a integra aceste funcţii sofisticate într-un sistem unitar. La alegerea (configurarea) unui sistem de AD&C este util să avem cunoştinţele de bază despre principiile de achiziţie de date.

Sisteme cu intrări analogice

Funcţia fundamentală a unui sistem cu intrări analogice este de a converti semnalele analogice într-un format numeric corespunzător. Această funcţie este realizată de convertorul ANALOG-NUMERIC (Analog to Digital - A/D), care transformă informaţia analogică în cod binar.

Pe lîngă convertorul A/D, pentru obţinerea unor performanţe optime este necesară utilizarea unor componente ca de exemplu: un amplificator, un circuit de eşantionare/memorare şi elemente de preprocesare a semnalului.

CONVERTOARE A/D - Azi, există un număr semnificativ de tipuri de convertoare A/D. Dintre acestea unele se diferenţiază prin faptul că sînt cele mai utilizate, ca de exemplu: aproximări succesive (Succesive Approximation Regulator - SAR), integrare (dual slope) şi convertoare paralele (FLASH). Deoarece convertoarele paralele sînt cele mai rapide, ele sînt şi cele mai scumpe. Majoritatea aplicaţiilor de AD&C, necesită o rezoluţie de cel puţin 12 biţi. Convertoarele de rezoluţie mai mare sînt nu numai mai scumpe dar şi mai lente, fapt previzibil de altfel. De aceea, este foarte important să definim cît mai precis necesităţile unei aplicaţii înainte de a ne decide asupra rezoluţiei folosite.

Un punct de pornire bun este TRADUCTORUL. Unele traductoare au o dinamică foarte mare. DINAMICA reprezintă diferenţa dintre nivelul maxim al semnalului de intrare şi cel mai mic nivel detectabil. Nu există o corelare precisă între precizia şi dinamica unui traductor. De exemplu un traductor cu precizia de 0,5% poate avea o dinamică de 80 decibeli (dB). Acest traductor are nevoie de un sistem cu o rezoluţie de 12 biţi. Pentru a menţine dinamica maximă, unele aplicaţii necesită o rezoluţie de 14 sau chiar 16 biţi. Amplificarea unui semnal de nivel mic, de 10 sau 100 de ori produce o creştere a rezoluţiei cu 3 respectiv 6 biţi.

Un sistem de 12 biţi împarte domeniul maxim al semnalului de intrare, în 4096 de părţi egale. Astfel, o parte reprezintă 1/4096 = 0,025% din întreg domeniul de lucru. O parte a unui sistem de 16 biţi, reprezintă 1/65536 = 0,0015% din întreg domeniul de lucru. De aceea, rezoluţia nu determină numai dinamica, dar limitează şi precizia întregului sistem. Pe de altă parte, mărind rezoluţia unui sistem, nu putem beneficia de acest avantaj decît dacă celelalte componente din sistem, ca de exemplu amplificatoarele sau circuitele de eşantionare/memorare nu acţionează ca factori limitatori în sistem.

Cînd variaţia unui semnal de intrare este mai mică decît rezoluţia MINIMĂ a sistemului, atunci acest eveniment va trece neobservat. De exemplu, la folosirea unui sistem de AD&C de 12 biţi (fără nici o preamplificare prealabilă), orice variaţie de semnal care nu depăşeşte 2,44 mV în domeniul de lucru maxim de 10 V, nu va putea fi detectată de sistemul de achiziţie. Pe de altă parte, dacă semnalul este mai întîi amplificat de 1000 de ori înainte de conversie, rezoluţia poate fi crescută la 2,44 mV (în absenţa zgomotului).

La viteze mai mari de 100 de eşantioane/secundă, convertorul bazat pe metoda aproximărilor succesive este cel mai indicat. De fapt, se pot obţine viteze de peste 100.000 eşantioane/secundă. Semnalul de intrare se compară cu semnale generate prin "GHICIRI" ponderate binare, pînă cînd cele două semnale devin identice. Este esenţial ca semnalul de intrare să rămînă NESCHIMBAT pe timpul comparărilor succesive, altfél, pot rezulta erori semnificative. Pentru a menţine semnalul de intrare CONSTANT, se foloseşte un circuit de eşantionare/memorare.

Cînd nu se lucrează la viteze mari, un convertor A/D bazat pe principiul INTEGRĂRII poate oferi rezoluţii de 12, 14 sau chiar 16 biţi la un preţ redus. Viteza de eşantionare nominală este cuprinsă între 3 şi 50 de conversii/secundă.

După cum îi spune şi numele, acest convertor MEDIAZĂ orice variaţie a semnalului de intrare pe timpul ciclului de conversie. Această proprietate filtrează în mod automat zgomotul de la intrare. Liniaritatea şi precizia pe întreg domeniul de lucru este în general mai bună decît la celelalte tipuri de convertoare A/D.

PRECIZIA este o mărime importantă a unui sistem de intrări analogice. Ea se defineşte ca eroarea totală la o anumită intrare. De exemplu, un sistem de AD&C care are o precizie de 0,05% pe întreg domeniul de 10 V, va genera o eroare, în cazul cel mai defavorabil, de 5 mV (10 V * 0,0005). Dacă precizia sistemului este de 0,1% pe un domeniu de +/-10 mV (de exemplu convertorul A/D pe domeniul de +/-10 V şi un amplificator cu

amplificare de 1.000) sistemul va genera în cazul cel mai defavorabil o eroare de 20 microvolţi (20 mV * 0,001).

În concluzie, aprecierea preciziei unui sistem de AD&C cere o analiză foarte atentă. Asiguraţi-vă că precizia dată se referă la întreg domeniul de lucru.

AMPLIFICATOARE - Amplitudinea semnalelor de intrare analogice poate varia în domenii foarte largi. Convertorul A/D necesită un semnal de amplitudine ridicată ca să funcţioneze optim. În majoritatea sistemelor se utilizează un amplificator pentru a ridica nivelul semnalelor mici la valorile cerute.

Ideal, amplificatorul de intrare ar trebui să aibă mai multe trepte de cîştig controlate prin soft. Acest dispozitiv este cunoscut sub numele de amplificator cu cîştig reglabil (Programmable Gain Amplifier - PGA). În realitate însă, compromisul între cost şi performanţe ne poate dicta folosirea unui PGA sau a unui amplificator cu cîştig modificabil MANUAL. Aici, "manual" înseamnă selecţia unui rezistor sau a unui ştrap pentru a fixa cîştigul pe placă.

Un etaj de intrare analogic se prezintă în figura 5. Să reţinem că amplificatorul de intrare nu este necesar la orice aplicaţie. După cum se vede acest circuit crează posibilitatea citirii doar a unui singur canal analogic.

O metodă care permite măsurarea mai multor canale ar fi folosirea unui convertor A/D şi a unui amplificator pentru fiecare canal de intrare. Oricum, există o metodă mult mai ieftină, descrisă mai jos.

MULTIPLEXOARE - Multiplexorul (MUX) arătat în figura 6 este o arhitectură de comutatoare electronice care permite ca un număr mare de canale să poată fi prelucrate folosind un singur amplificator şi un singur convertor A/D. Softul poate controla aceste comutatoare anlogice pentru a selecta oricare din canalul dorit în vederea procesării la un anumit moment dat de timp. Această metodă oferă o reducere considerabilă a costului, faţă de utilizarea amplificatoarelor şi convertoarelor A/D, separat pentru fiecare canal în parte.

Deoarece amplificatorul şi convertorul A/D sînt folosite în comun, viteza de achiziţie se va reduce. Ca primă aproximaţie, viteza amplificatorului şi a convertorului A/D va fi divizată la numărul canalelor de intrare gestionate.

RANDAMENTUL EŞANTIONĂRILOR se defineşte prin viteza unui canal înmulţită cu numărul de canale.

EŞANTIONARE ŞI MEMORARE (SAMPLE & HOLD - S/H) - În general, semnalul analogic de intrare poate varia în timp. După cum am văzut, convertoarele A/D bazate pe aproximaţii succesive sînt condiţionate de cerinţa ca semnalul de intrare să NU se modifice pe perioada ciclului de conversie. Funcţia circuitului de eşantionare şi memorare (S/H) este de a eşantiona valoarea prezentă la intrare înainte de începerea conversiei. Acest nivel de semnal este ţinut constant, în ciuda modificării semnalului de la intrare, pînă cînd ciclul de conversie s-a terminat. Această proprietate asigură conversia precisă a semnalelor de înaltă frecvenţă.

MULTIPLEXAREA ÎN TIMP - În sistemul descris în figura 6, canalele folosesc în comun amplificatorul, circuitul S/H şi convertorul A/D. Utilizatorul selectează rata de eşantionare dorită în funcţie de aplicaţia dată. Dacă fiecare canal trebuie citit de R ori pe secundă, atunci MUX-ul va trebui să eşantioneze cu o rată egală cu n * R (unde n este numărul canalelor citite). Deci, circuitul S/H şi convertorul A/D trebuie să fie atît de rapide, încît să execute un ciclu de conversie într-un timp mai mic decît 1/n*R secunde. Trebuie să avem grijă, deci, să nu fim induşi în eroare de specificaţiile de viteză a diferitelor componente din sistem.

TIMPUL DE CONVERSIE - defineşte numai viteza convertorului A/D, care reprezintă numai o parte

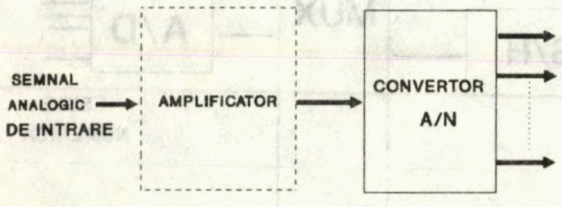


Fig. 5. Etaj de intrare analogic

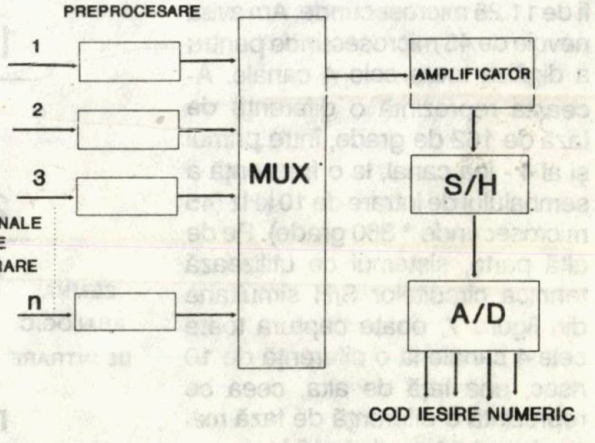


Fig. 6. Subsistem analogic de intrare

din timpul total necesar pentru a măsura un canal dat. Deci, pentru a afla viteza reală a întregului sistem trebuie să ştim atît rata de eşantionare per canal sau randamentul eşantionării cît şi condiţiile în care au fost specificate (de exemplu randamentul eşantionării este funcţie de cîştigul amplificatorului). Ideal, toate canalele de intrare vor fi citite în acelaşi moment de timp, adică la fiecare 1/R secunde. Oricum, multiplexarea în timp generează o "ASIMETRIE" sau o diferenţă de timp între citirile canalelor. Dacă MUX-ul, S/H-ul şi convertorul A/D sînt suficient de rapide, atunci putem avea impresia că intrările sînt citite simultan. Unele aplicaţii sînt foarte sensibile la asimetria de timp, ca de exemplu măsurarea puterii electrice instantanee (I*U) sau măsurarea poziţiei relative a componentelor mecanice. Chiar dacă folosim cele mai rapide convertoare A/D, există încă multe aplicaţii care nu pot "tolera" diferenţa de timp între citiri rezultată în urma eşantionărilor secvenţiale. În aplicaţii critice, tehnica folosirii circuitelor de eşantionare/memorare simultane, poate reduce asimetria de timp de 100 pînă la 1000 de ori. Această arhitectură este ideală pentru aplicaţii în care relaţia dintre fază şi timp a multiplelor canale de intrare este critică în tehnica investigării folosite. De exemplu, dacă sistemul din figura 6 ar eşantiona 4 intrări analogice cu un randament de 89.000 eşantioane/secundă, timpul mort între două conversii ar fi de 11,25 microsecunde. Am avea nevoie de 45 microsecunde pentru a digitiza toate cele 4 canale. Aceasta reprezintă o diferenţă de fază de 162 de grade, între primul şi al 4 - lea canal, la o frecvenţă a semnalului de intrare de 10 kHz (45 microsecunde * 360 grade). Pe de altă parte, sistemul ce utilizează tehnica circuitelor S/H simultane din figura 7, poate captura toate cele 4 canale la o diferenţă de 10 nsec, una faţă de alta, ceea ce reprezintă o diferenţă de fază mai mică de 0,04 grade la 10 kHz.

Această tehnică este recomandată mai ales acolo unde trebuie calculate funcţii de corelaţie. De exemplu, analiza vorbirii, măsurarea puterii în sistemele trifazate, analiza semnalelor geofizice şi testarea automată a echipamentelor (ATE).

PREPROCESAREA SEMNALELOR - Chiar dacă se utilizează componente de înaltă calitate, este de dorit o preprocesare a semnalelor de la intrare. Preprocesarea, poate fi de două tipuri: ACTIVĂ - care include o amplificare şi o izolare electrică, sau PASIVĂ - care include o divizare de tensiune, conversie curent-tensiune, filtrare şi suprimarea brumului de reţea. Amplitudinea maximă a semnalului care poate fi aplicată la intrarea unui amplificator sau convertor A/D este de obicei +/-10 V. Divizori rezistivi se folosesc pentru a transforma orice nivel de tensiune în acest domeniu acceptat de valori.

Astfel, se poate realiza monitorizarea unui semnal de 48 V sau 480V. Este important a se lua în considerare că multiplexorul şi celelalte componente electronice se pot avaria, dacă se conectează la intrarea acestora semnale ce depăşesc 15 V. Prin adăugarea la circuitele de preprocesare a unor dispozitive de limitare, cum ar fi diode Zenner sau MOV-uri (Metal Oxid Varistor), se poate asigura o protecţie faţă de posibilele vîrfuri de tensiune de la intrare.

Cîteodată este de dorit să se preamplifice semnalele de nivel mic (1mV la 1V) în afara cutiei în care se găseşte convertorul A/D pentru a obţine un raport semnal-zgomot maxim. Una din aceste metode este "EMIŢĂTORUL PE DOUĂ FIRE". Emiţătoarele realizează nu numai o amplificare a semnalelor de intrare ci pot face şi izolarea electrică, liniarizarea şi conversia în curent la o valoare relativ mare (tipic 4 la 20 mA) Transmiterea semnalelor în curent se poate face la distanţe de pînă la 1500m fără diminuarea semnificativă a preciziei de lucru, în timp ce semnalele de tensiune sînt atenuate foarte repede de rezistenţa cablului de interconectare. Semnalele de curent se pot comuta simplu în tensiune prin intermediul unui rezistor. Valorile cuprinse între 250 şi 500 de ohmi sînt cele mai uzuale, producînd căderi de tensiune de 5 pînă la 10V corespunzător valorilor de curenţi între 4 şi 20 mA)

Dintre toate categoriile de preprocesare, filtrarea este cea mai dorită, cea mai utilizată şi cea mai prost înţeleasă. Practic, filtrarea este folosită pentru a separa semnalele dorite de cele nedorite. Semnalele nedorite includ: zgomot, brum de reţea, interferenţă electromagnetică captată de la

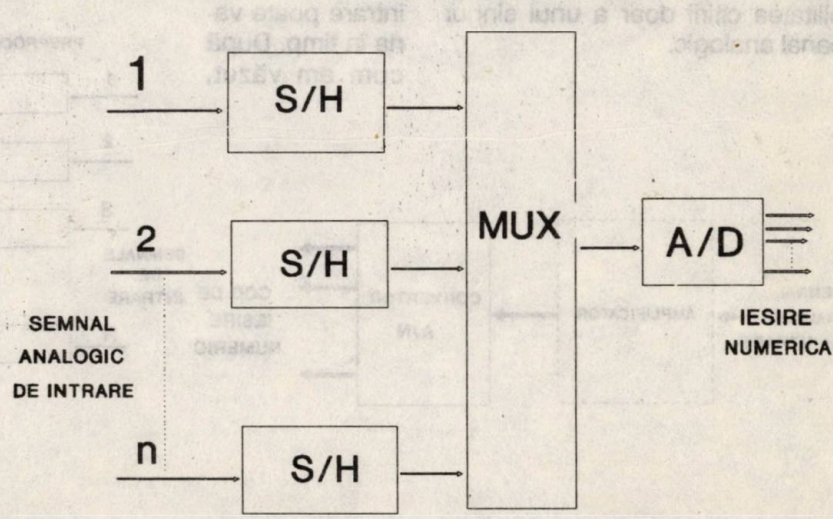


Fig. 7. Sistem cu circuite S/H simultane

staţiile de radio - TV şi semnalele cu frecvenţa peste 1/2 din frecvenţa de eşantionare. În general, se foloseşte un filtru trece-jos (LOW PASS FILTER) pentru a atenua aceste surse nedorite de eroare. Cînd la intrarea convertorului A/D, apar semnale cu frecvenţa peste 1/2 din frecvenţa de eşantionare folosită, ia naştere un fenomen numit FALSIFICARE (ALIASING). Aceasta se manifestă prin generarea de semnale "FALSE" în domeniul de frecvenţă interesat, care nu pot fi distinse de informaţia reală. Astfel, se pot produce erori foarte mari în interpretarea datelor.

Semnale în mod comun (SINGLE - ENDED) şi semnale diferenţiale (DIFFERENTIAL)

Semnalele analogice pot fi configurate ca intrări de mod comun sau diferenţiale. Intrările de mod comun se definesc toate în raport cu o singură MASĂ comună. Aceste semnale se pot utiliza doar în cazul în care căderea de tensiune pe traseul de masă este mică. Problemele încep să apară cînd această cădere de tensiune creşte. Aceasta generează bucle de masă (scurgeri de curenţi prin conductorul de masă) ceea ce poate genera tensiuni de eroare la intrările convertorului A/D. Principalul avantaj al acestei metode este costul scăzut per canal. Este necesar doar un singur multiplexor pentru a trata fiecare canal în parte.

Intrările diferenţiale folosesc două fire: intrarea inversoare (-) şi intrarea neinversoare (+) a amplificatorului, pentru a se conecta la sursa de semnal. Astfel, orice semnal indus printr-o buclă de masă, va apare ca un semnal de mod comun şi va fi rejectat de proprietăţile diferenţiale ale amplificatorului.

Conexiunile diferenţiale pot reduce considerabil efectele buclelor de masă, dar ele necesită două multiplexoare per canal. Astfel, un sistem de 32 de canale în mod comun, va avea 16 intrări diferenţiale.

În unele aplicaţii se utilizează şi aşa numitele conexiuni PSEUDO - DIFERENŢIALE. Aceasta este de fapt, o conexiune în mod comun, în care una din intrări este conectată la masa comună a semnalelor de intrare. Astfel, acest canal, măsoară tensiunea indusă în bucla de masă, care poate fi corectată prin soft. Această tehnică este folositoare acolo unde toate semnalele de intrare folosesc ca referinţă acelaşi potenţial de masă.

Amplificatoare de instrumentaţie - INSTRUMENTATION AMPLIFIERS (IA) - Un amplificator de instrumentaţie este un bloc cu intrări diferenţiale cu impedanţă de intrare foarte mare la ambele intrări, + şi -. Proprietăţile de rejecţie de mod comun, atenuează efectele buclelor de masă, brumul de reţea precum şi semnalele de eroare induse prin zgomot. Astfel IA, este folositor în special pentru a mări semnalele de nivel mic. Dacă IA are cîştigul programabil prin soft, el se numeşte PGIA - PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER (amplificator de instrumentaţie cu cîştig programabil).

Ideal, impedanţa de intrare, rejecţia de mod comun şi lărgimea de bandă a amplificatoarelor ar trebui să fie infinit de mare. Aceasta ar însemna ca valorile curentului de intrare şi tensiunii de offset să fie NULE. Cu alte cuvinte, acest lucru ar însemna ca sursa de semnal să nu fie influenţată de circuitul de măsurare.

În realitate, însă, amplificatoarele au impedanţe de intrare, curenţi de intrare şi tensiuni de offset de valori finite.

TENSIUNEA DE OFFSET - (Vos) a unui amplificator se referă la tensiunea de ieşire a amplificatorului cînd la intrare se aplică o tensiune NULĂ (intrările sînt scurtcircuitate). Practic, Vos este tensiunea ce ar trebui amplificată la intrările +/- ale amplificatorului pentru ca ieşirea lui să fie NULĂ. Existenţa Vos, se datorează micilor diferenţe existente între componentele ce alcătuiesc circuitul de intrare al unui amplificator.

Cu toate că, majoritatea amplificatoarelor sînt prevăzute cu reglaje pentru eliminarea offsetului, aceasta se face însă sacrificînd alţi parametri. De exemplu eliminarea offsetului prin reglaj, generează deseori creşterea driftului de offset (Vos se modifică cu temperatura) şi alte efecte nedorite. Vos poate fi compensată şi prin soft. În cele mai multe cazuri, CURENŢII DE INTRARE dau mai multă bătaie de cap. Pentru denumirea curenţilor de intrare se folosesc doi termeni: CURENTUL DE BIAS (BIAS CURRENT - Ib) şi CURENTUL DE OFFSET (OFFSET CURRENT - Ios). Ib se referă la curentul care intră sau iese în sau din terminalul + sau - al amplificatorului. Curentul Ios este diferenţa dintre curenţii de bias al intrărilor + şi -. În principiu, diferenţa dintre cei doi termeni este importantă, pentru că Ios poate fi mult mai mic decît Ib. Aceşti curenţi interacţionează cu impedanţa sursei de semnal, producînd o tensiune de offset adiţională. Dacă impedanţa sursei este balansată (BALANCED), adică ea este identică la ambele intrări, + şi -, se generează eroare numai datorită lui Ios.

Este esenţială existenţa unui conductor extern între intrările amplificatorului şi masa sursei de alimentare, a cărei rezistenţă trebuie să fie atît de mică încît tensiunea de offset rezultată, Ib*Rs să nu înrăutăţească performanţele amplificatorului. În situaţia extremă, cînd intrările amplificatorului sînt lăsate LIBER, adică nu există conductor de masă, amplificatorul ajunge într-o stare în care NU poate fi utilizat.

REJECTIA DE MOD COMUN - Abilitatea amplificatorului cu intrare diferenţială de a discrimina între semnalul de intrare diferenţial şi semnalul de mod comun (semnalul nedorit) pentru amplificare diferenţiată se defineşte prin raportul rejecţiei de mod comun (Common Mode Rejection Ratio - CMRR) în decibeli. Pentru un amplificator dat, raportul dintre amplificarea sa în mod comun (Gcm) şi amplificarea sa diferenţială (Gdiff) este:

CMRR = $20\log_{10}$(Gcm/Gdiff) exprimat în decibeli.

Amplificarea în mod comun se determină prin măsurarea variaţiei tensiunii de ieşire care rezultă datorită unei schimbări la intrarea de mod comun.

Gcm = DVout/DVcm

Înlocuind această valoare în ecuaţia CMRR, obţinem:

$DVout = DVcm * Gdiff * 10^{(-CMRR/20)}$

Ca o regulă generală, CMRR-ul unui amplificator se îmbunătăţeşte odată cu creşterea amplificării sale în buclă închisă. Într-un sistem complet de AD&C semnalul de ieşire din amplificator (incluzînd erorile datorate valorii finite a CMRR) ajunge la intrarea convertorului A/D. Acesta nu poate discerne între părţile utile şi eronate ale semnalului de la intrare.

Semnificativă este numai relaţia care există între amplitudinea semnalului de eroare şi sensibilitatea convertorului A/D (de exemplu 1 LSB - bitul cel mai puţin semnificativ). Dacă eroarea depăşeşte 1 LSB, convertorul A/D, răspunde. De aceea este mult mai de dorit ca să exprimăm CMRR în unităţi LSB. Acest lucru se face prin divizarea tensiunii de eroare de mod comun (DVout) la SENSIBILITATEA convertorului.

Sensibilitatea convertorului A/D, se calculează prin divizarea întregului domeniu de lucru (FSR - FULL SCALE RANGE), la rezoluţie

- Eroarea de mod comun (în unităţi LSB) = (DVcm * Gdiff * $10^{(CMRR/20)}$)/FSR/Rezoluţie

Tabela ce urmează, ne arată relaţia dintre eroarea CMRR, exprimată în decibeli şi unităţi LSB, pentru un sistem ipotetic. Pentru această comparaţie s-a considerat un sistem cu o rezoluţie de 12 biţi pe un domeniu de 10V (0 la 10V sau +/-5V). Un semnal de mod comun de 10V este aplicat la intrările scurtcircuitate ale sistemului. Împărţind eroarea de mod comun (exprimată în unităţi LSB) la tensiunea de mod comun, rezultă o informaţie utilă cu privire la performanţele întregului sistem. Observaţi că, o creştere a CMRR la un cîştig dat, realizează o creştere a performanţelor sistemului. Oricum, o creştere relativ mică a CMRR, obţinută ca urmare a unei creşteri a cîştigului, produce o diminuare a preciziei sistemului.

Cîştig de semnal	CMRR	EroareCM absolută	EroareCM/VCM
1	80dB	0.4LSB	0.04LSB/Volt
10	90dB	1.3LSB	0.13LSB/Volt
100	100dB	4.1LSB	0.41LSB/Volt
1000	110dB	13LSB	1.3LSB/Volt

IEŞIRI ANALOGICE - În majoritatea sistemelor, ieşirile analogice sînt necesare. Aceste semnale se folosesc ca semnale de reacţie la controlul în buclă închisă. Domeniul de variaţie a acestor semnale poate fi: +/-5V, +/-10V, 0-10V şi 4-20mA. Ieşirile analogice, se realizează cu convertoare numeric/analogice - D/A, care pot lucra în sarcină, cu curenţi de pînă la 5 sau chiar 10 mA. Aceasta nu reprezintă o limitare, pentru că majoritatea aplicaţiilor, necesită comanda unor sarcini ce au impedanţe ridicate. În cazul în care se comandă sarcini cu impedanţe de valori mici, ca de exemplu: electrovalve, lămpi indicatoare, motoare electrice etc., se folosesc amplificatoare de putere sau generatoare de curent constant de valori mari. Acestea se vor ataşa la sistem, deoarece majoritatea sistemelor de AD&C NU includ etaje de putere.

INTRĂRI ŞI IEŞIRI NUMERICE - Cele mai multe sisteme de AD&C pot accepta şi genera semnale TTL (0 la 5V). În cazul aplicaţiilor ce folosesc alte nivele discrete de tensiune, sînt necesare interfeţe speciale. Pentru a comanda dispozitive ca: bobine, motoare şi relee sînt necesare nivele mai mari de tensiune şi curenţi.

Pentru a facilita conectarea cablelor de la traductori la sistemul de AD&C există o varietate de tipuri de BORNIERE (TERMINATION PANELS). Aceste borniere, conţin şuruburi adiţionale de cuplare la proces pentru modulele de preprocesare a semnalelor, indicatoare luminoase - LED pentru afişarea stărilor canalelor urmărite, divizoare de tensiune şi circuite de izolare electrică. Astfel, se pot monitoriza atît nivele înalte de curent continuu, cît şi circuite de curent alternativ.

INTRĂRI ŞI IEŞIRI DE NUMĂRARE - Există o varietate largă de aplicaţii de numărare. Alte aplicaţii cer ca anumite dispozitive, să fie pornite sau oprite la momente de timp foarte precise. Toate aceste funcţii, pot fi rezolvate cu circuite de tip numărare/temporizare (COUNTER/TIMER - C/T), care sînt optimizate pentru aplicaţii ce includ măsurarea de frecvenţe şi generarea bazelor de timp. Numărătoarele se caracterizează prin numărul evenimentelor de intrare care se pot urmări şi de frecvenţa lor maximă de la intrare. Majoritatea sistemelor folosesc numărătoare de 16 biţi care pot înregistra impulsuri la frecvenţa maximă de 8 MHz. Capacitatea maximă a numărătoarelor de 16 biţi

este de 65536 de evenimente. Aceste numărătoare sînt toate independente unele faţă de celelalte şi pot fi utilizate pentru numărarea de evenimente, măsurarea sau divizarea de frecvenţe. Generatoarele de impulsuri, permit programarea prin soft atît a duratei impulsurilor cît şi a frecvenţei acestora. Un asemenea generator de impulsuri, se foloseşte, de obicei pentru a genera o bază de timp precisă, folosită pentru achiziţia datelor la momente precise de timp. Numărătoarele sistemelor de AD&C, acceptă nivele TTL şi pot număra pe un număr de canale predefinit prin soft. Numărarea poate fi startată de la o valoare iniţială definită şi, de asemenea, numărătorul poate fi configurat pentru a se reseta la această valoare după ce valoarea sa a fost citită.

Intern, numărătorul îşi decrementează valoarea la fiecare impuls sosit la intrare. Dacă un numărător de 16 biţi depăşeşte valoarea de 65535, are loc depăşirea capacităţii sale, generîndu-se un semnal de depăşire. Acest semnal poate fi folosit la activarea unor evenimente externe.

Măsurarea frecvenţelor folosind numărătoare, poate fi realizată pe mai multe căi, dependent de tipul aplicaţiei. Dacă frecvenţa necunoscută este un semnal TTL, el se poate aplica direct la numărător. Semnalele analogice cu o amplitudine de cel puţin 100 mV, pot fi convertite în semnale TTL, cu ajutorul modulelor de TRIGGERARE (ALARM/TRIGGER). Divizoarele de tensiune folosind rezistenţe ŞI/SAU diode Zenner sau opto - izolatori, pot fi utilizate la divizarea cu aceste trepte,a semnalelor de intrare de nivele mari. În cazul în care se utilizează circuite de preprocesare, este necesară o analiză în ceea ce priveşte limitarea de viteză.

Există două opţiuni distincte pentru măsurarea frecvenţelor joase şi înalte. Prima metodă determină perioada semnalului necunoscut, comparîndu-l cu perioada cunoscută a semnalului unui generator de ceas. Avantajul acestei metode este că permite o măsurare a semnalelor de joasă frecvenţă cu o rezoluţie mare şi minimizează timpul necesar folosit pentru măsurare. De obicei, metoda se foloseşte sub frecvenţe de 10 Hz.

A doua metodă, numără ciclii semnalului necunoscut într-un interval de timp FIXAT. Avantajul acestei tehnici este că permite măsurarea semnalelor peste 8 MHz. Este relativ uşor să se implementeze un algoritm de auto - calibrare ce optimizează rezoluţia peste un domeniu foarte larg de frecvenţe.

Tehnici software

Softul este cel care face viabil un sistem de AD&C. Un hard ales are o valoare nesemnificativă fără soft. Achiziţia de date pe PC-uri a fost concepută în aşa fel, încît utilizatorii au posibilitatea de a scrie programe speciale pentru achiziţie de date, numărare, afişare şi control, în limbaje de nivel înalt. Cînd un soft este livrat cu un sistem de AD&C, acest lucru se face în aşa fel încît anumite taskuri să fie cît mai simplu de realizat pentru utilizator.

În general, există, pe PC-uri 3 clase de soft pentru sistemele de AD&C:

- 1) programe de învăţare şi dezvoltare
- 2) librării de subrutine, ce conţin funcţii
- 3) pachete de aplicaţii cu meniuri, livrate la " CHEIE "

Programele de la punctele 1 şi 2 sînt proiectate pentru a permite utilizatorilor posibilitatea de a-şi concepe şi de a scrie singuri o aplicaţie soft, UNICĂ. Acestea includ, de obicei "DRIVERE" care reprezintă interfaţa cu hardul de I/E. Aceste pachete, fac ca scrierea programelor în limbaje de nivel înalt, ca de exemplu BASIC, C, TURBO PASCAL sau ASSYST să fie foarte uşoară. Acest tip de programare este foarte flexibil şi util pentru scopuri generale.

Programele de la punctul 3 sînt făcute pentru a pune în funcţiune IMEDIAT întregul sistem de AD&C, de obicei fără necesitatea de a mai scrie alte module program. Oricum, unele dintre aceste pachete oferă utilizatorilor posibilitatea de a modifica softul la cerinţele lor. În mod normal, acest tip de produs este oferit pentru o aplicaţie specifică. De aceea, aceste pachete sînt mai puţin structurate şi mai puţin flexibile ca cele din alte clase. Tot aici, putem aminti numeroase colecţii de SOFT GENERIC, care sînt programe scrise de firme independente pentru produsele hard ale altor firme. Unele dintre cele mai cunoscute includ: LABTECH NOTEBOOK, HYPERSIGNAL-WORKSTATION şi SNAP. Acestea, precum şi multe altele, pot realiza pe lîngă culegerea, analiza, plotarea datelor şi funcţii de control.

Multe, dacă nu chiar toate, dintre aplicaţiile de AD&C depind de execuţia în timp a operaţiilor de intrare/ieşire. Cînd viteza ŞI/SAU timpul este critic, este necesară considerarea a 3 tehnici de lucru soft: POLLING (baleierea), prin ÎNTRERUPERI şi DMA.

POLLING-ul este cea mai simplă metodă de a detecta îndeplinirea unei condiţii şi luarea unei decizii. Aceasta implică o buclă de program care conţine toţi algoritmii de măsurare, analiză şi luare a deciziilor. Programul de AD&C testează periodic ceasul sistemului sau o intrare conectată la un element extern pentru a analiza dacă nu s-a produs o schimbare. Cînd apare o schimbare, programul eşantionează toate intrările şi memorează valoarea acestora într-un TAMPON. Un tampon este o zonă de memorie care conţine valorile ce reprezintă intrările la un anumit moment dat de timp. Tamponul poate fi generat în RAM, pe disc sau pe alte tipuri de memorii. De

fiecare dată cînd programul " SIMTE " o tranziţie a ceasului de sistem - un TICK - intrările sînt eşantionate şi convertite şi o nouă valoare se adaugă în tampon.

În acest mod, un PC/AT 286 poate achiziţiona date cu o rată de aproximativ 18kHz, depinzînd de ceasul procesului. Pe de altă parte, PC-ul este proiectat în aşa fel încît trebuie avută în vedere o toleranţă de timp (astfel, o toleranţă de timp de 12 microsecunde la un PC este un lucru obişnuit. De asemenea, PC-ul este încontinuu ocupat cînd execută bucla de polling şi astfel nu mai poate executa alte taskuri. Cînd un program nu poate "tolera" aceste restricţii, va trebui să aplicăm tehnica întreruperilor.

ÎNTRERUPERILE permit, pe lîngă controlul precis al timpului într-un sistem de AD&C, şi executarea altor taskuri concomitent cu supravegherea procesului. Sistemele multitasking sînt cunoscute şi sub numele de sisteme "FAŢĂ/SPATE" - FOREGROUND/BACKGROUND. Una din metodele de a pune taskul de achiziţie de date în "spate" este de a-l lega de o rutină de întrerupere. Semnalul de ceas sau un semnal extern, în loc de a fi urmărit continuu prin polling, este folosit pentru a genera o întrerupere în PC. La apariţia unei întreruperi, calculatorul îşi suspendă activitatea curentă şi execută o rutină de întrerupere de serviciu (ISR - INTERRUPT SERVICE ROUTINE). ISR-ul, în acest caz poate fi un program scurt care eşantionează toate intrările şi le memorează în tampon. Calculatorul poate executa şi alte operaţii în "faţă" în timp ce culege datele în "spate". La apariţia unui TICK sau a unei întreruperi externe, calculatorul va reveni în punctul din program de unde a plecat.

Timpul de răspuns al sistemului de întreruperi este mult mai lung decît timpul unei bucle de polling bine concepute. Acest lucru se datorează faptului că, mecanismul de întreruperi, în majoritatea calculatoarelor, implică o cantitate însemnată de soft. Viteza unui PC, în modul de lucru cu întreruperi, este de circa 4 kHz. De asemenea, pot apărea complicaţii la scrierea unui soft de întreruperi. În mai toate cazurile, programatorul, trebuie să fie pregătit să scrie programe în limbaj de asamblare. În contrast, majoritatea sistemelor ce utilizează metoda polling pot fi scrise în limbaje de nivel înalt. Întreruperile sînt avantajoase în situaţii în care rata de achiziţie de date este lentă, precizia în timp nu e prioritară şi aplicaţia "din spate" este cea importantă. Cînd timpul necesar pentru a executa o rutină ISR este mic în comparaţie cu rata de apariţie a întreruperilor în sistem, atunci această tehnică poate da rezultate excelente. Aceste considerente trebuiesc bine cîntărite înainte de a alege între un sistem bazat pe polling şi altul bazat pe întreruperi.

Metoda accesului direct la memorie (DIRECT MEMORY ACCES - DMA) este o tehnică hard care asigură cea mai mare viteză de transfer a datelor, din sau în RAM. Tehnicile DMA utilizează echipament hard mai scump şi pot realiza citiri şi scrieri de date la momente precise de timp, fără impunerea unor restricţii semnificative în ceea ce priveşte microprocesorul în execuţia taskurilor.

De exemplu, PC-ul lucrînd în mod DMA poate citi sau scrie orice combinaţii de date (analogice, numerice sau de numărare) din sau în RAM la o rată maximă de 360 kocteţi/secundă. Această performanţă este obţinută prin "FURT DE CICLU - CYCLE STEALING",adică prin "oprirea" pentru perioade scurte de timp a UC, pentru a permite executarea în acest timp a unor transferuri directe în sau din memorie - fără participarea unităţii centrale. Timpul necesar achitării unei cereri DMA este mult mai mic decît cel necesar executării unui ISR, ceea ce permite realizarea comutării rapide între aplicaţii.

(ing. Kallo Tibor)

Cîteva adrese

La solicitarea unor corespondenţi ai revistei noastre, vă prezentăm adresele unor sedii principale ale cîtorva dintre cele mai importante firme din domeniul instrumentaţiei de laborator.

Burr Brown - Burr Brown Overseas Marketing Office, 1 Milfield House, Woodshots Meadow, Watford, Hertfordshire, England, WD1 8YX

Keithley - Keithley Instruments GmbH, Heiglhofstrasse 5, 8000 München 70

Metrabyte - 440 Myles Standish Boulevard, Taunton, MA 02780

Philips - Philips Export B.V., I&E Export, Building HVW3, P.O.Box 218, 5600 MD Eindhoven, Netherland;

- reprezentanţă în România: Institutul Politehnic Philips Service, P.O.Box 1/285, str. Polizu 1, Bucureşti, dl. Stoian

Siemens - Siemens AG, Bereich Automatisierungstechnik, Geschäftsgebiet Messtechnik, Postfach 4848, D-8500 Nürnberg 1

Rohde & Schwarz - Rohde & Schwarz GmbH, Sonnleithnergasse 20, A 1100 Wien

Ce este IEEE 488?

Folosirea calculatoarelor personale în tehnica măsurătorilor PCI (Personal Computer Instrumentation)

Tehnica măsurătorilor de laborator se află în faţa unei schimbări esenţiale. În timp ce înainte consta în achiziţionarea şi afişarea unor date pur analogice, lucru care se făcea cu aparate de măsură dedicate pentru efectuarea măsurării unor valori, azi predomină tehnicile digitale. Datorită existenţei unor componente electronice ieftine şi compacte aparatele de măsură devin din ce în ce mai "inteligente" şi permit achiziţionarea şi interpretarea datelor rezultate din măsurători prin metode noi. Hotărîtor pentru noua dezvoltare este integrarea crescîndă a calculatoarelor personale în activitatea de laborator, PC-urile permiţînd controlarea unui ansamblu de aparate de măsură conectate la calculator. Astăzi sistemele de achiziţie a datelor din măsurători se împart astfel:

- - sisteme independente
  - — aparate de măsură simple utilizate individual (stand-alone)
  - —aparate de măsură care permit comunicaţia cu sisteme de măsură, dar se folosesc în principal în regim independent
- -PC-uri dedicate pentru măsurători avînd plachete specializate sau care permit racordarea unor module de extensie pentru achiziţionarea datelor de măsură
- -sisteme bus pentru laborator construite după proceduri byte-seriale sau bit-seriale şi care permit interconectarea de aparate de măsură cu calculatoare avînd o interfaţă adecvată

Avantajele utilizării calculatoarelor personale la efectuarea unor măsurători în laboratoare (în proiectare, controlul calităţii, medicină, cercetare, învăţămînt) sînt:

- -utilizarea uşoară şi care permite o achiziţie de date uşor controlabilă şi interpretabilă
- -programarea uşoară a procedurii de măsură
- -repetabilitatea măsurătorilor, care permite un volum mare de măsurători în condiţii identice
- -existenţa unor programe standard
- -o interfaţă utilizator unitară (display, tastatură, mouse) pentru toate aparatele de măsură din configuraţie
- -preluarea uşoară a datelor obţinute din măsurători

Datorită avantajelor pe care le prezintă, se estimează pînă în '93 o creştere anuală de peste 20% pe plan mondial, în domeniul aparatelor ce permit efectuarea măsurătorilor cu PC-uri.

Interfaţa IEC 625 (IEEE-488.1 respectiv 2, HPIB, GPIB)

(denumirile distincte desemnează în principiu acelaşi concept; există însă diferenţe de detaliu)

Caracteristici:

- nr. de aparate ce se pot interconecta: 15
- mod de conectare: bus pasiv, funcţiile de interfaţă fiind realizate în cadrul fiecărui aparat în parte
- cablul de racord: 16 fire active şi 8 de masă, ecranate în comun, avînd o lungime maximă de 20m, respectiv 2m pentru fiecare aparat
- mufa de racord: 25 pini
- metoda de transmisie a datelor: byte-serială, bit paralelă, asincronă
- viteza: maxim 1 Mbyte/sec la lungime limitată de bus; la lungime maximă 200...500 kbyte/sec
- lungime maximă mesaj: 10...20 caractere
- nivel semnal: TTL
- posibilităţi de adresare: 1 byte de adresă ce permite adresarea a maxim 31 de emiţători ("vorbitori" - talker) şi 31 de receptori ("ascultători" - listener)

Interfaţa specificată în IEC 625 este de uz general şi permite racordarea aparaturii de măsură automate (programabile) la calculator, într-un sistem bus. Comunicaţia între aparate au loc sub formă de mesaje. Distingem:

- -mesaje de interfaţă - care sînt legate de condiţiile de funcţionare a sistemului
- -mesaje dependente de aparat - legate de funcţionarea unui anumit aparat
- Din punctul de vedere al funcţiilor interfeţei aparatura se împarte în

- emiţător (talker). Aparatul care poate emite este adresabil în mod selectiv cu un mesaj de interfaţă şi , în starea activă (adresat) poate trimite un mesaj dependent de aparat pe bus. În cadrul sistemului doar un singur aparat poate fi emiţător activ la un moment dat şi mesajul aflat pe bus poate proveni doar de la acesta.
- receptor (listener). Acest tip de aparat poate fi adresat selectiv printr-un mesaj de interfaţă şi în stare activă (adresat) poate recepţiona de pe bus un mesaj de aparat. Pot fi active simultan mai multe aparate receptoare.

■ -de comandă (controller). Acest tip de aparat (de obicei calculatorul) este capabil să adreseze selectiv celelalte aparate, să asigneze funcţionarea de receptor şi emiţător şi să transmită şi alte mesaje de interfaţă referitoare la toate sau doar anumite aparate din sistem. Aparatele avînd doar funcţie de controller nu recepţionează şi emit mesaje dependente de aparat.

Funcţia îndeplinită de aparate în sistemul de interfaţă poate fi dedicat sau modificabil. Un aparat poate avea funcţii diferite în momente diferite.

Fiecare aparat are o adresă dedicată, care poate fi de emisie, de recepţie sau amîndouă. Două aparate pot avea aceeaşi adresă de recepţie, dacă trebuie să recepţioneze aceleaşi date, dar nu pot fi două aparate cu aceeaşi adresă de emisie.

Configuraţia minimă este de un emiţător şi un receptor, fără aparat de comandă. De exemplu un voltmetru digital racordat la o imprimantă care tipăreşte rezultatele măsurătorilor semiautomat.

Structura de bus

Avantajul structurii este posibilitatea de a extinde, reduce sau reconfigura uşor sistemul, dar există dezavantajul necesităţii de adresare şi comandă mai complicate, înlăturate de programele existente.

În figura 1 se prezintă structura bus-ului IEC

Pe cele 8 linii ale busului de date se transmit mesajele de interfaţă. Mesajele pot fi de adresă, dată de program, rezultate de măsură sau diferite instrucţiuni şi pot proveni de la orice aparat cu funcţie de control sau emiţător. Denumirea liniilor este Data Input/Output, **DIO1...DIO8.**

Rolul celor 3 linii de comandă a transferului de date este de a comanda transferurile pe liniile DIO. Vitezele de transfer ale diferitelor aparate diferă semnificativ şi în cazul transferului de mesaje pe mai mulţi bytes este necesară o asigurare că emisia are loc atunci cînd se poate recepţiona şi să nu fie mai rapidă decît cel mai lent aparat de recepţie. Aceste trei linii asigură dialogul (handshake - "strîngerea de mînă") asincron. Notaţia liniilor este:

DAV - data valid - pe nivel L înseamnă că datele sînt valide şi se pot recepţiona

NRFD - not ready for data - pe H indică prezenţa datelor pregătite pentru recepţie

NDAC - no data accepted - pe H indică preluarea datelor de către receptor

Aceste semnale sînt date de către interfeţele aparatelor ce iau parte la comunicaţie şi nu de către aparatul cu funcţii de control.

Cele 5 linii de comenzi generale au fiecare un rol specific în comanda transferului de date:

IFC - interface clear - iniţializează interfeţele. În sistemele cu mai multe aparate cu rol de control poate fi dat doar de cel cu prioritatea cea mai mare

ATN - attention - determină modul în care se interpretează informaţia de pe bus-ul de date. Pe nivel

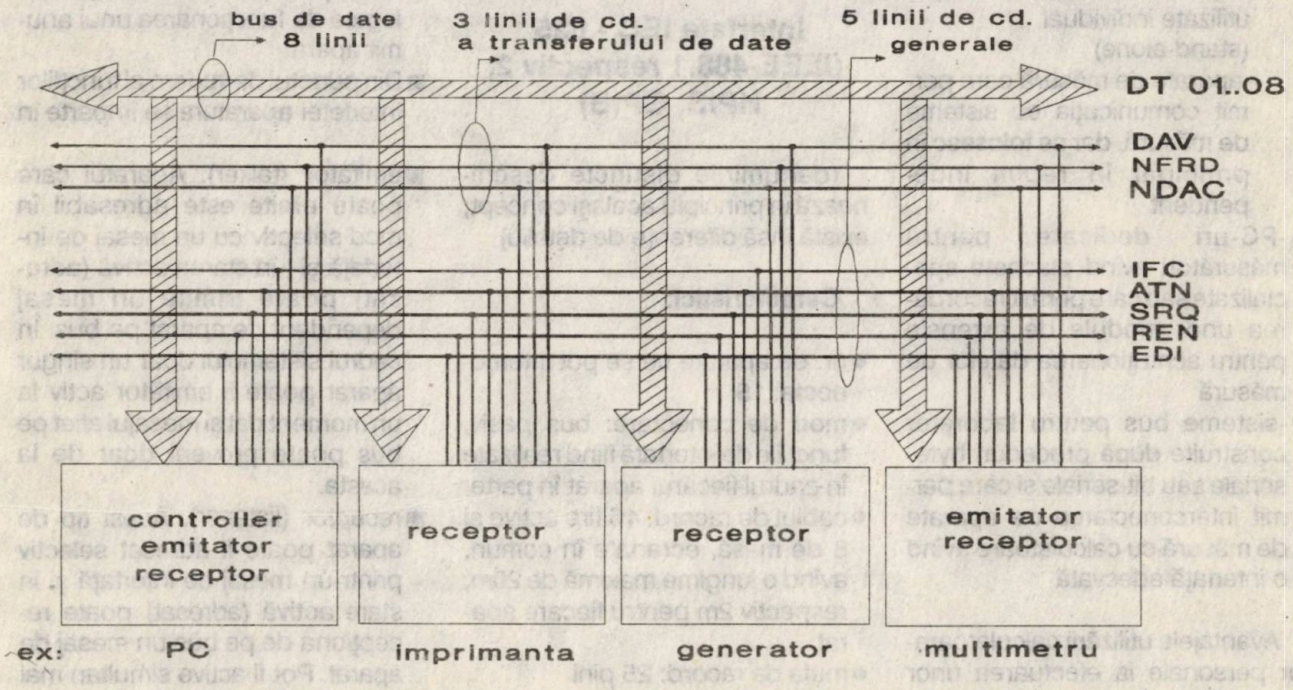


Fig. 1. Structura bus-ului IEC

L avem un mesaj de interfaţă iar pe H un mesaj de aparat.

SRQ - service request - pe această linie emiţătorii pot cere de la aparatul de comandă (controller) întreruperea activităţii în curs.

REN - remote enable - această linie este folosită de aparatul de comandă (controller) împreună cu alte linii, pentru comutarea unui aparat din sistem în regim de telecomandă

EOI - end or identify - această linie este folosită pentru desemnarea ultimului element dintr-un mesaj din mai mulţi bytes şi de asemenea pentru interogarea (POLL) liniilor de către controller.

Mufele folosite sînt de două tipuri:

- în **standard IEC** - mufe IEC 48b (secretariat) 80A trapezoidală, cu 25 pini cilindrici (Fig. 2).
- în **standard IEEE** - mufă Amphenol sau Cinch series 57, MICRO-RIBBON de 24 pini (Fig. 3).

Cablurile au cîte două mufe la fiecare capăt, una tată şi una mamă, ceea ce permite poziţionarea lor încălecată (piggy-back) multiplă.

Diagrama de semnale de dialog (Fig. 4).

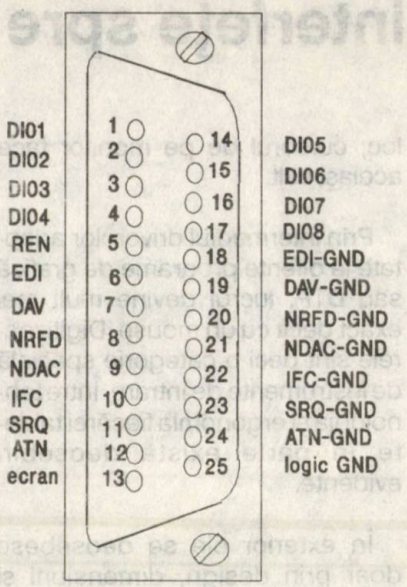


Fig. 2. Mufă standard IEC

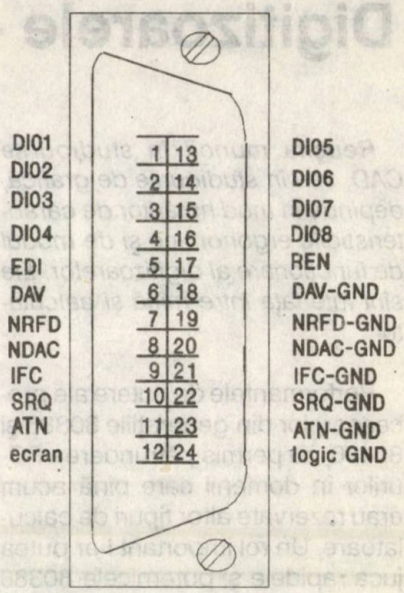


Fig. 3. Mufă standard IEEE

La pornirea transferului de date emiţătorul activ aşteaptă pînă ce receptorii activi sînt gata pentru recepţia unui byte de date. Datele sînt emise pe bus, După semnalarea existenţei unor date valide pe bus se aşteaptă ca ele să fie acceptate de toţi receptorii. După ce datele au fost acceptate emiţătorul emite următorul byte pe bus şi aşteaptă din nou ca receptorii să devină pregătiţi pentru primirea datelor. Cînd acest lucru are loc emiţătorul indică faptul că datele sînt valide şi recepţia se repetă pînă la terminarea transferului.

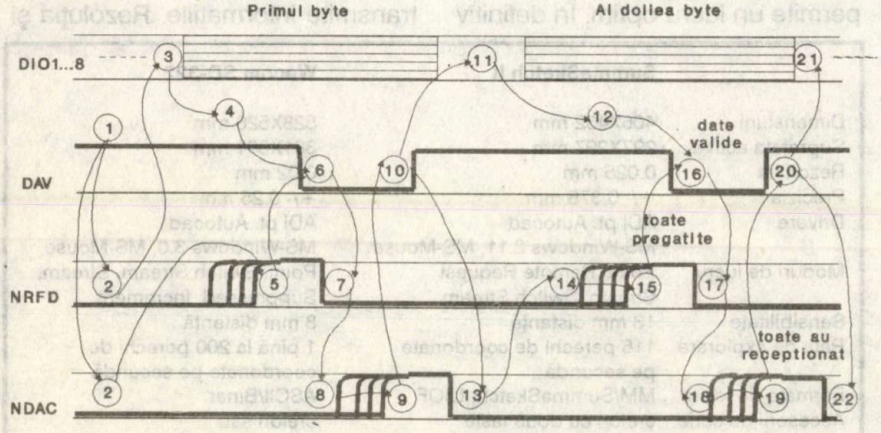


Fig. 4. Diagrama de semnale de dialog

La punerea sub tensiune controllerul de sistem (un PC) este totdeauna controllerul activ şi controlează toate tranzacţiile pe bus. Controllerul de sistem poate trece controlul unui alt aparat din sistem care devine controller activ, dar menţine totdeauna controlul ultim prin liniile IFC şi REN prin care poate prelua controlul oricînd doreşte.

Există o multitudine de firme care furnizează interfeţe pentru bus IEEE-488. Astfel firma **Burr Brown** oferă PCI-801K, PCI-803W pentru bus de PC-XT/AT şi PCI-802K, PCI-804W pentru MCA, **Metrabyte** MBC-488, IE-488 pentru bus PC-XT/AT şi respectiv IBM-GPIB adapter şi CEC-PC488 pentru MCA, **ines** -ieee 488 pentru bus NI, AT, PS2 etc.

Aceste firme oferă şi softul necesar pentru o activitate de laborator complexă; Amintim aici ASYST /ASYSTANT GPIB, LOTUS MEASURE, GURU. Lab Windows, DaDisp488 etc., toate avînd, de regulă, interfeţe pentru limbajele de programare cele mai răspîndite (Basic, Pascal, C).

(ing. Darvas Attila)

Digitizoarele - interfeţe spre grafică şi artă

Reuşita muncii în studiourile CAD, sau în studiourile de grafică, depinde în mod hotărîtor de caracteristicile ergonomice şi de modul de funcţionare al digitizoarelor. Ele sînt interfaţa între mînă şi calculator.

Performanţele de putere ale procesoarelor din generaţiile 80386 şi 80486, au permis pătrunderea PC-urilor în domenii care pînă acum erau rezervate altor tipuri de calculatoare. Un rol important l-ar putea juca rapidele şi puternicele 80386 în cazul locurilor de muncă individuale din domeniile DTP şi CAD. Achiziţia unui PC din această categorie nu depinde numai de puterea de calcul a CPU ci şi de puterea mediului soft şi de echipamentele de intrare/ieşire de calitate, compatibile 100% IBM. Instrumentul la care nici un utilizator DTP, şi nici un grafician, nu poate renunţa este digitizorul. El mijloceşte cea mai ergonomică posibilă colectare şi introducere a ilustraţiilor complexe şi a construcţiilor. Cei care nu sînt utilizatori DTP sînt de părere adesea că digitizoarele de departe lucrează după acelaşi principiu de funcţionare ca un mouse. Valabilitatea acestei presupuneri se limitează doar la faptul că un digitizor poate emula cele mai răspîndite tipuri de mouse. Deosebirea fundamentală între modul de lucru relativ al unui mouse şi modul de lucru absolut al unui digitizor este altul. În modul de lucru relativ, denumit şi mod mouse, poziţia "lupei" sau a "creionului" pe digitizor nu corespunde cu punctul corespunzător al cursorului de pe ecran. Indiferent din ce poziţie de pe digitizor se reia desenul, cursorul este pus în mişcare de la ultima sa poziţie. În modul de lucru absolut, coordonatele elementului de deservire (lupă sau creion) de pe digitizor corespund cu poziţia actuală a cursorului de pe monitor. Dacă lupa sau creionul sînt ridicate de pe digitizor şi puse într-un alt loc, cursorul de pe monitor face acelaşi salt.

Prin intermediul driverelor adaptate la diferite programe de grafică sau DTP, lucrul devine mult mai exact decît cu un mouse. Digitizoarele sînt deci o categorie specială de instrumente de intrare. Între tehnologia şi ergonomia fiecărei tablete în parte există deosebiri evidente.

În exterior ele se deosebesc doar prin design, dimensiuni şi greutate. Sistemul folosit de ele decide în care domeniu poate fi folosit eficient fiecare digitizor în parte. Pentru exemplificare am comparat modul de utilizare în DTP şi CAD pentru două tipuri de digitizoare: Summasketch II, produs de Summagraphics, şi SD-332, produs de Wacom.

Cumpărătorul potenţial trebuie să se gîndească bine dacă preţul scăzut este un criteriu în alegerea sa. Digitizorul Summasketch este mai ieftin, dar este în mod evident şi mai puţin performant tehnic decît digitizorul Wacom. Datorită tehnicii sale fără cablu şi aşa-zisei tehnologii "Give and Receive Resonance", Wacom deţine un avans inovativ. Tocmai în domeniile în care creativitatea utilizatorului trebuie să se desfăşoare nestingherită, Wacom permite un lucru optim. În definitiv nu degeaba sînt folosite digitizoarele Wacom în studiourile de animaţie ale industriei americane de film.

În domeniul CAD se utilizează mai ales digitizoare Summasketch. Ele se caracterizează printr-o flexibilitate mare şi o deservire simplă. Cele mai importante drivere oferite sînt cele pentru MS-Windows, ADI pentru Autocad şi Autosketch, şi un emulator de mouse. Ca accesoriu de serie se livrează un creion şi o lupă cu patru taste (la cerere poate fi livrată şi o lupă cu 16 taste). Digitizorul Summasketch se conectează prin cablu la interfaţa serială.

Manualul de prezentare oferă, într-o formă concentrată, toate informaţiile necesare unei instalări cu succes. Tehnologia folosită de Summasketch funcţionează după principiul electromagnetic, numit şi de inducţie. Undele transmise de elementele de introducere cu o anumită frecvenţă sînt recepţionate, controlate şi retransmise calculatorului de către digitizor.

Digitizorul Summagraphics lucrează, din acest punct de vedere, cu totul altfel decît digitizorul Wacom. Lupa sau creionul au doar rol de emiţător, tabela fiind cea care recepţionează, comandă şi retransmite informaţiile. Rezoluţia şi

	SummaSketch II	Wacom SD-322
Dimensiuni	406X432 mm	528X526 mm
Suprafaţa activă	297X297 mm	381X381 mm
Rezoluţia	0.025 mm	0.02 mm
Precizia	+/- 0.375 mm	+/- 0.25 mm
Drivere	ADI pt. Autocad MS-Windows 2.11, MS-Mouse	ADI pt. Autocad MS-Windows 3.0, MS-Mouse
Moduri de lucru	Point, Remote Request Stream, Switch Stream	Point, Switch Stream, Stream, Suppressed, Increment
Sensibilitate	13 mm distanţă	8 mm distanţă
Rata de explorare	116 perechi de coordonate pe secundă	1 pînă la 200 perechi de coordonate pe secundă
Formate de ieşire	MM/SummaSketch, UIOF	ASCII/Binar
Accesorii de serie	creion cu două taste lupă cu patru taste	creion sau lupă cu patru taste
Preţ	cca. 1500 DM	cca. 4100 DM

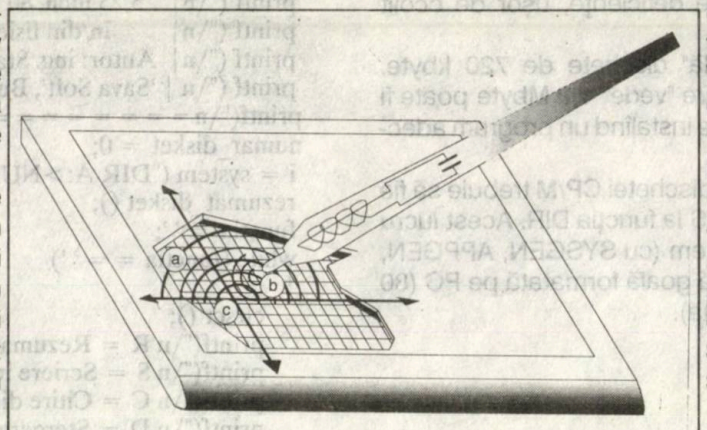
pasul pot fi comandate prin calculator cu o secvenţă de comenzi. Şi alegerea modurilor de lucru (Point, Remote, Request, Stream sau Switch Stream) se poate face cu ajutorul unui utilitar aflat pe dischetа din furnitura de livrare. Suprafaţa activă de lucru este de 297 mm X 297 mm (deci format DIN A4). Rezoluţia este de 10, 20 sau 40 de rînduri/mm şi poate fi aleasă în funcţie de cerinţele fiecărui monitor în parte. Precizia senzorilor din cîmpul activ de lucru este de +- 0,375 mm.

Digitizorul Summasketch este cu ceva mai subţire şi mai uşor decît digitizorul Wacom. Ergonomia este bună. În interiorul domeniului de lucru activ cablul lupei nu are efecte de îngrădire. La folosirea creionului cablul poate deranja însă.

Digitizorul Wacom se prezintă diferit deja prin forma sa exterioară şi prin greutate. Wacom A4+ este cu 4 kg mai greu şi mult mai robust decît concurentul său Summasketch II.

Digitizorul Wacom foloseşte tehnologia electromagnetică Give-and-Receive-Resonance, care elimină necesitatea cablului. Utilizatorul poate lucra în domeniul de lucru fără restricţii şi cu precizie mare. Pentru a obţine acest spaţiu liber de mişcare, Wacom foloseşte o tehnologie care schimbă la fiecare 20 de microsecunde distribuţia rolurilor dintre digitizor şi creion, (sau lupă), ca emiţător/receptor.

În rolul de emiţător digitizorul emite unde electromagnetice de o anumită frecvenţă. Ele sînt recepţionate de un circuit de rezonanţă paralel aflat în creion. Tensiunea indusă în bobina circuitului este înmagazinată într-un condensator legat în paralel cu bobina. Energia înmagazinată aici este folosită de creion pentru a retransmite undele pe o altă frecvenţă digitizorului. Digitizorul poate localiza deci, în modul de lucru receptor, undele transmise de creion şi-i poate stabili poziţia. Procedeul acesta de comutare permanentă emiţător/receptor este o particularitate a digitizorului Wacom care permite lucrul fără cablu. O altă caracteristică care derivă din acest procedeu este posibilitatea de a produce linii mai groase sau mai subţiri în funcţie de cît de tare este apăsat creionul pe suprafaţa digitizorului. Presiunea exercitată asupra creionului modifică frecvenţa de rezonanţă a circuitului paralel din creion şi digitizorul poate sesiza dacă s-a apăsat creionul mai tare sau mai încet. În acest mod poate fi simulată şi folosirea unei pensule (la unele programe de pictură), digitizorul Wacom putînd fi folosit şi în scopuri pur artistice.



Prezentare schematică a modului de funcţionare al digitizorului Wacom:
a) în mod emisie, digitizorul emite unde electromagnetice
b) creionul emite pe o altă frecvenţă spre digitizor
c) reţeaua fină de conductori a digitizorului înregistrează şi localizează poziţia şi tăria cu care este apăsat creionul

Digitizorul este conectat printr-un conector SUB-D cu nouă pini la interfaţa serială. Domeniul de lucru activ în formatul DIN A4 este de 30 cm X 30 cm. În domeniul activ de lucru al modelului SD322 a fost integrată o placă detaşabilă din sticlă acrilică. Sub această placă pot fi aşezate folii de meniuri, suporturi sau planuri de comutaţie, ceea ce simplifică simţitor transferul desenelor şi prelucrarea lor cu diferite programe. Driverele oferite sînt: GSL, ADI pentru Autocad, un driver pentru Windows 3.0 şi un driver de mouse Microsoft. Configurarea digitizorului poate fi făcută prin intermediul blocului de microîntrerupătoare 24DIP de pe spatele tabletei, care permite setul de comenzi Bid-Pad-Two proprii Wacom sau setul de comenzi mouse Microsoft. Dischetele din furnitura de livrare conţin pe lîngă drivere şi cîteva utilitare importante. Cu ele pot fi testate, de ex., conexiunile.

Cu ajutorul softului se poate alege modul de lucru relativ sau absolut şi în plus poate fi aleasă unitatea de măsură, rata de transfer şi rezoluţia. Digitizorul Wacom atinge o rezoluţie de 0,02 mm şi recepţionează semnale emise de la 8 mm de lupă şi de la 4 mm de creion. Precizia este de maxim +- 0,25 mm. Digitizorul SD-322 poate folosi unul din modurile de operare: Point, Stream sau Switch Stream.

În domeniul CAD ambele tipuri de digitizoare au performanţe comparabile. Din punct de vedere al preţului digitizorul Summasketch este mai avantajos. Digitizorul Wacom are ca avantaje prelucrarea precisă şi robusteţea. În cazul în care digitizorul este folosit cu programe DTP sau de pictură, tehnica mai avansată a digitizorului Wacom nu poate decît să convingă. Pe suprafaţa de lucru utilizatorul poate lucra în mod asemănător pictării sau desenării libere pe un carton. Elementele de introducere fără cablu prezintă caracteristici ergonomice perfecte. Caracteristicile de bază ale digitizorului Wacom sînt calitatea înaltă de prelucrare şi tehnica optimizată.

(PC+Technik 1/91, Reiner König)

Să nu uităm CP/M-ul...

Programul DX

Doream să aduc de pe dischete CP/M în harddisk-ul PC-ului niște fișiere. Nu reușeam să fac acest lucru cu ajutorul legăturii seriale din mai multe motive. Întrucît dischetele erau de 5 1/4 inch (80 piste, dublă față, 720 kbyte) provenind de la CUB-Z, mi-am propus și am realizat un program pe PC care să lucreze cu astfel de dischete (citire, scriere, rezumat, ștergere).

Acest program s-a dovedit ulterior mult mai comod decît transmisia serială (cu Kermit, Move-it etc.), devenind o componentă deosebit de utilă în condițiile prelucrărilor distribuite, pe CUB-Z și PC.

Programul are și unele deficiențe, ușor de ocolit însă:

- PC-ul trebuie să "vadă" dischete de 720 kbyte. Precizez că orice AT care "vede" 1,2 Mbyte poate fi făcut "să vadă" 720 kbyte instalînd un program adecvat.
- primele două piste ale dischetei CP/M trebuie să fie recunoscute de MS-DOS la funcția DIR. Acest lucru se realizează punînd sistem (cu SYSGEN, APPGEN, 5GEN) de pe o dischetă goală formatată pe PC (80 piste, 9 sectoare pe pistă).

ing. Sava Stan, Buz{u

```
#include <dos.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <io.h>
#include <fcntl.h>

#define CITIRE   1
#define SCRIERE 2

void beep (void);
void bloc_dx ();
void rezumat_disket (void);
void afisrez (void);

short numar_disket;
short abscisa, ordonata;
short i,j,k,l,numar_fisiere, inceput;
short nf, nrf;
unsigned char c, user = 0;
unsigned char buf [2048];
unsigned char blocuri [360];
unsigned char fisiere [128][12];
unsigned char sort [128];
unsigned char i1,i2;
unsigned char art [80];
unsigned char numefis[30], numf [12];
unsigned short numar_sectoare [128];
unsigned short block [351];
unsigned short nb, nrbl, nrsect;
char nrext;
long lung, kk;

void main ()
{
  unsigned char functia;
  clrscr ();
  printf("\n= = = = = = = = = = = = = = = = = = = =");
  printf ("\n|          Program DX                |");
  printf ("\n| Conversie fisiere disketa CP/M     |");
  printf ("\n|   5,25 inch, 80 piste, 2 fete      |");
  printf ("\n|      in/din fisiere MS-DOS         |");
  printf ("\n| Autor: ing. Stan Sava 02-sep-90    |");
  printf ("\n | 'Sava Soft', Buzau, tel. 974/12760|");
  printf("\n= = = = = = = = = = = = = = = = = = = =");
  numar_disket = 0;
  i = system ("DIR A:>NULL");
  rezumat_disket ();
  functia = ' ';
  while (functia = = ' ')
  {
     clrscr ();
     printf("\n R = Rezumat disketa");
     printf("\n S = Scriere in disketa");
     printf("\n C = Citire din disketa");
     printf("\n D = Stergere fisier in disketa");
     printf("\n T = Terminare program \n\n");
     abscisa = wherex ();
     ordonata = wherey ();
     c = ' ';
     while (c = = ' ')
     {
        gotoxy (abscisa, ordonata);
        printf ("Functie (R,S,C,D,T): ");
        gotoxy (abscisa + 19, ordonata);
        if (( c = getch ()) = = '\0')
        {
          getch ();
          c = ' ';
          continue;
        }
        putch (c);
        switch (c)
        {
          case 'R':
          case 'r':
                afisrez ();
                if (getch () = = '\0') getch();
                c = '1';
                functia = ' ';
                break;
          case 'C':
          case 'c':
                afisrez ();
                if (numar_fisiere = = 0)
```

```
        {
            printf ("\Disc gol");
            beep ();
            exit (0);
        }
        printf ("\nNumar fisier:");
        abscisa = wherex ();
        ordonata = wherey ();
        nrf = 0;
        while (nrf = = 0)
        {
            gotoxy (abscisa, ordonata);
            printf ("   ");
            gotoxy (abscisa, ordonata);
            buf [0] = 4;
            if (numar_fisiere < 100)
               buf [0] = 3;
            if (numar_fisiere < 10)
               buf [0] = 2;
            cgets (buf);
            if (buf [1] = = 0)
            {
               nrf = -1;
               c = '1';
               break;
            }
            nrf = atoi (buf+2);
            if (nrf < 1 || nrf > numar_fisiere)
            {
               beep ();
               nrf = 0;
            }
        }
        if (nrf = = -1)
        {
            functia = ' ';
            break;
        }
        i1 = sort [nrf-1];
        for (i=0; i<8;
            numefis [i] =fisiere [i1] [i++]);
        numefis [8] = '.';
        for (i=8; i<11;
            numefis[i+1] =fisiere [i1 [i++]);
        numefis [12] = '\0';
        if ((nf = open (numefis,
            O_BINARY|O_RDWR)) < 5)
            nf = _creat (numefis, FA_ARCH);
        k = numar_sectoare [i1] + 1;
        for (i=0; i<360; i++)
        {
            if (blocuri [i] != (i1+1) )
               continue;
    bloc_dx (numar_disket, i, buf, CITIRE);
    for (j=0; j<16; j++)
    {
        k--;
        if (k = = 0) break;
        l = write (nf, &buf [j*128], 128l);
    }
}
close (nf);
functia = ' ';
c = '1';
break;

case 'S':
case 's':
    clrscr ();
    printf ("\nNume fisier:");
    abscisa = wherex ();
    ordonata = wherey ();
    nrf = 0;
    while (nrf = = 0)
    {
        functia = '1';
        gotoxy (abscisa, ordonata);
        printf ("          ");
        gotoxy (abscisa, ordonata);
        buf [0] = 13;
        cgets (buf);
        if (buf [1] = = '\0')
        {
            functia = ' ';
            c = '1';
            break;
        }
        strcpy (numefis, buf+2);
        for (i=0; i<11; numf [i++] = ' ');
        strupr (numefis);
        j = 0;
        for (i=0; i<12; i++)
        {
            if (numefis [i] = = '\0') break;
            if (numefis [i] = = '.')
            {
               j = 8;
               continue;
            }
            numf [j++] = numefis [i];
        }
        for (i=0; i<numar_fisiere; i++);
        {
            for (j=0; j<11; j++)
               if (fisiere [i] [j] != numf [j]) break;
            if (j = = 11)
            {
               printf ("\nFisier deja prezent");
               beep ();
               functia = ' ';
               c = '1';
               break;
            }
    }
    if (functia = = ' ') break;
    if ((nf = open (numefis,
        O_BINARY|O_RDONLY)) < 5)
    {
        printf ("\nFisier %s absent",numefis);
        beep ();
        functia = ' ';
        c = '1';
        break;
    }
    if (functia != ' ') nrf = 1;
}
if (functia = = ' ') break;
lung = filelength (nf);
nrsect = (lung+127) / 128;
nrbl = (nrsect+15) / 16;
j = 0;
```

```
    for (i=2; i<351; i++)
    if (blocuri [i] == '\0') j++;
    if (j < nrbl)
    {
        printf ("\nFisier prea mare");
        close (nf);
        functia = ' ';
        c = '1';
        break;
    }
    nb = 0;
    for (i=2; i<351; i++)
    {
        if (blocuri [i] != '\0') continue;
        block [nb++] = i;
        if (nb == nrbl) break;
    }
    nrext = -1;
    nb = 0;
    for (j=0; j<2; j++)
    {
        if (nb >= nrbl) break;
        bloc_dx (numar_disket, j, buf, CITIRE);
        for (i=0; i<64; i++)
        {
            if (nb >= nrbl) break;
            inceput = i * 32;
            if (buf [inceput] == user) continue;
            buf [inceput] = user;
            nrext++;
            for (l=0; l<11; l++)
                buf [inceput+l+1] = numf [l];
            buf [inceput+12] = nrext;
            buf [inceput+13] =
                buf [inceput+14] = '\0';
            k = nrsect - 128 * nrext;
            if (k > 128) k = 128;
            buf [inceput+15] = k;
            for (l=0; l<8; l++)
            {
                if (nb >= nrbl)
                buf [inceput+16+l+1] =
                    buf [inceput+17+l+1] = '\0';
            else
            {
                buf [inceput+16+l+1] =
                    block [nb] % 256;
                buf [inceput+17+l+1] =
                    block [nb++] / 256;
            }
        }
    }
    bloc_dx (numar_disket, j, buf, SCRIERE);
}
for (i=0; i<nrbl; i++)
{
    kk = 2048;
    if (kk > lung) kk = lung;
    lung = kk;
    read (nf, buf, kk);
    while (kk < 2048) buf [kk++] = 0x1a;
    bloc_dx (numar_disket, block [i], buf, SCRIERE);
}
close (nf);
rezumat_disket ();
    functia = ' ';
    c = '1';
    break;
case 'D':
case 'd':
        afisrez ();
        printf ("\nNumar fisier:");
        abscisa = wherex ();
        ordonata = wherey ();
        nrf = 0;
        while (nrf == 0)
        {
            gotoxy (abscisa, ordonata);
            printf ("  ");
            gotoxy (abscisa, ordonata);
            buf [0] = 4;
            if (numar_fisiere < 100) buf [0] = 3;
            if (numar_fisiere < 100) buf [0] = 2;
            cgets (buf);
            if (buf [1] == 0)
            {
                nrf = -1;
                break;
            }
            nrf = atoi (buf+2);
            if (nrf < 1 || nrf > numar_fisiere)
            {
                beep ();
                nrf = 0;
            }
        }
        if (nrf == -1)
        {
            functia = ' ';
            c = '1';
            break;
        }
        j = sort [nrf - 1];
        for (i=0; i<11; numf[i] = fisiere [j] [i++]);
        k = 0;
        for (j=0; j<2; j++)
        {
            bloc_dx (numar_disket, j, buf, CITIRE);
            for (i=0; i<64; i++)
            {
                inceput = i*32;
                if (buf [inceput] != user) continue;
                buf [inceput+9] &= 0x7f;
                for (l=0; l<11; l++)
                {
                    if (numf [l] != buf [inceput+l+1])
                        break;
                }
                if (l == 11)
                {
                    k = 1;
                    buf [inceput] = 0xe5;
                }
            }
            if (k == 1)
            bloc_dx (numar_disket, j, buf, SCRIERE);
        }
        rezumat_disket ();
        functia = ' ';
```

```
            c = '1';
            break;
        case 'T':
        case 't':
            exit (0);
        }
        }
        }
}

void afisrez ()
{
    clrscr ();
    k = (numar_fisiere + 2) / 3;
    for (i = 0; i < k; i + +)
    {
        for (i2 = 0; i2 < 3; i2 + +)
        {
            l = k * i2 + i;
            if (l > = numar_fisiere) continue;
            i1 = sort [l];
            for (j = 0; j < 8; numf[j] = fisiere [i1] [j + +]);
            for(j = 8; j < 11; numf[j + 1] = fisiere [i1][j + +]);
            numf [8] = '.';
            numf [12] = '\0';
            if (i2 = = 0) printf ("\n");
            printf ("%02d.%s %4d", l + 1, numf,
                numar_sectoare [i1]);
            if (i2 ! = 2) printf ("      ");
        }
    if (i = = 24)
        if (getch () = = 0) getch ();
    }
}

void rezumat_disket (void)
{
    unsigned short cc;

    for (i = 0; i < 360; blocuri[i + +] = '\0');
    numar_fisiere = 0;
    for (j = 0; j < 2; j + +)
    {
        bloc_dx (numar_disket, j, buf, CITIRE);
        for (i = 0; i < 64; i + +)
        {
            inceput = i* 32;
            if (buf [inceput] ! = user) continue;
            buf [inceput + 9] & = 0x7f;
            buf [inceput + 10] & = 0x7f;
            buf [inceput + 11] & = 0x7f;
            for (k = 0; k < numar_fisiere; k + +);
            {
                for (l = 0; l < 11; l + +);
                if (buf [inceput + l + 1] ! = fisiere [k] [l])
                    break;
                if (l = = 11) break;
            }
            if (k = = numar_fisiere)
            {
                numar_fisiere + +;
                for (l = 0; l < 11; l + +)
                    fisiere [k] [l] = buf [inceput + l + 1];
                numar_sectoare [k] = 0;
            }
            numar_sectoare [k] + = buf [inceput + 15];
            for (l = 16; l < 32; l + = 2)
            {
                cc = buf [inceput + l] +
                    buf [inceput + l + 1] * 256 ;
                if (cc = = 0) break;
                blocuri [cc] = k + 1;
            }
        }
    }
    for (i = 0; i < numar_fisiere; sort[i] = i + +);
    for (i = 0; i < numar_fisiere; i + +)
    for (j = i + 1; j < numar_fisiere; j + +)
    {
        i1 = sort [i];
        i2 = sort [j];
        for (l = 0; l < i1; l + +)
        {
            if (fisiere [i1] [l] < fisiere [i2] [l]) break;
            if (fisiere [i1] [l] = = fisiere [i2] [l]) continue;
            sort [i] = i2;
            sort [j] = i1;
            break;
        }
    }
}

void beep (void)
{
    printf ("%c", 7);
}

void
bloc_dx (numar_disc, numar_bloc, buffer, operatia)
short numar_bloc, numar_disc,
operatia;
unsigned char buffer [];
{
    unsigned char blocuri [36] =
        { 0, 2, 4, 6, 8, 1, 3, 5, 7,
        9,11,13,15,17,10,12,14,16, 18,
        20,22,24,26,19,21,23,25,27,29,
        31,33,35,28,30,32,34};
    short j1, j2, k, adresa_sector;

    j1 = numar_bloc / 9 + 1;
    j1 * = 36;
    j2 = numar_bloc % 9;
    for (k = 0; k < 4; k + +)
    {
        adresa_sector = blocuri [j2*4 + k] + j1;
        if (operatia = = CITIRE)
        {
            disable ();
            absread (numar_disc, 1,
                adresa_sector, buffer + k*512);
            enable ();
        }
        if (operatia = = SCRIERE)
            abswrite (numar_disc, 1,
                adresa_sector, buffer + k*512);
    }
}
```

Sfaturi practice pentru întreţinerea harddisk-urilor

Discurile hard sînt foarte utile atît timp cît funcţionează. Acest articol vă dă cîteva sfaturi pentru depanarea lor.

Odată şi odată în viaţa oricărui utilizator vine un moment în care doreşte să încheie lucrarea, care l-a costat multă osteneală în ultimele luni. O intenţie lăudabilă, numai că se pare că în chiar acel moment PC-ul a intrat în grevă. Apar mesaje de eroare, sub forma unor cifre nedefinite, sau mesaje de genul:
"Drive not ready" sau
"Boot Disk failure" care produc panică în rîndul utilizatorilor. Cel mai tîrziu în acel moment va apare părerea de rău, că ultima salvare a fost făcută cu multă vreme în urmă. Dacă nici "mîngîierea" calculatorului şi nici alte metode promiţătoare nu ajută la nimic, atunci doar cineva care ştie ce este defect mai poate ajuta. Trebuie ştiut însă un lucru: în 90% din cazuri nimic nu este defect.

Prima diagnoză

Calculatorul d-voastră nu mai recunoaşte dintr-o dată harddisk-ul, sau nu mai vrea să extragă datele memorate ? Ar trebui să verificaţi că într-adevăr hardisk-ul este stricat, şi deci că pana este hard.

- Puteţi stabili acest lucru fiind atenţi la zgomotul pe care calculatorul d-voastră îl face atunci cînd este pus în funcţiune. Harddisk-urile produc zgomote specifice la pornire. Dacă zumzetul obişnuit al discului nu se face auzit, atunci cel mai probabil este că sînteţi ghinionist şi că discul este într-adevăr defect.

- Observaţi cu atenţie ecranul în timpul autotestului de la pornirea PC-ului. Dacă eroarea este recunoscută de către sistem atunci ea va fi anunţată printr-un ţiuit sau prin apariţia unui cod de eroare pe ecran, pe care îl puteţi descifra folosind un tabel de coduri de eroare.
- Dacă autotestul dovedeşte în mod clar existenţa unui defect pe controller-ul harddisk-ului (unitatea de comandă a harddisk-ului), atunci acesta va trebui depanat într-o unitate specializată. Celelalte probleme pot fi rezolvate de obicei prin programe speciale.

Salvarea datelor pe dischete

Un harddisk defect îşi anunţă din timp iminenţa prăbuşirii. Dacă observaţi în timpul lucrului erori tot mai dese la citirea sau scrierea pe disc, este bine să salvaţi cît mai repede datele pe dischete, metodă denumită "backup pe dischete".

Cel mai bine este să efectuaţi un backup complet pe dischete. Pentru aceasta aveţi la dispoziţie comenzile DOS "Backup" şi "Restore" cît şi o serie de programe specializate cum ar fi "Corefast" al firmei Core International sau "Fastback Plus" al firmei Fifth Generation Systems.

Dacă harddisk-ul şi-a făcut datoria fără să crîcnească şi fără zgomote suspecte pînă în "ziua Z", atunci problema va putea fi rezolvată cu programe ajutătoare. Pentru aceasta aveţi nevoie de o dischetă sistem sau de una de diagnosticare. Nu este util să ţineţi programele ajutătoare pe harddisk, căci cum le-aţi putea folosi dacă nu puteţi porni discul ?

Primul ajutor

Introduceţi discheta sistem în unitatea A: şi porniţi din nou calculatorul. Probabil va urma acelaşi mesaj de eroare şi veţi fi solicitat să apăsaţi o tastă. Urmaţi această solicitare şi aşteptaţi pînă cînd calculatorul se va anunţa cu ecranul obişnuit de start (aşa-numitul prompter sistem). Încercaţi să accesaţi acum harddisk-ul. Introduceţi următoarea comandă
"C: <Enter>". Dacă acest lucru funcţionează fără probleme puteţi să vă destindeţi pentru moment, datele încă mai pot fi salvate.

- Afişaţi apoi conţinutul harddisk-ului cu comanda **"DIR"**

- Verificaţi că afişarea fişierelor se efectuează ca şi pînă acum şi fiţi atenţi în mod special dacă interpretorul de comenzi al sistemului de operare MS-DOS **"COMMAND.COM"** apare în această listă. Dacă acesta lipseşte schimbaţi din nou unitatea pe A: şi transferaţi sistemul pe harddisk cu comanda:
"SYS C: <Enter>".

- Apoi copiaţi fişierul "COMMAND.COM" de pe discheta sistem pe harddisk

- Scoateţi discheta din unitate şi porniţi din nou calculatorul.

Ultima salvare

Dacă încercarea de a schimba unitatea pe C: eşuează cu mesajul de eroare: "Invalid drive specification", atunci executaţi programul "FDISK" de pe discheta sistem. FDISK este un program standard MS-DOS, destinat administrării harddisk-ului. Mare atenţie însă, cu acest program un necunoscător poate distruge foarte uşor datele!

- La apariţia meniului introductiv al lui **FDISK** alegeţi punctul 4 al meniului, referitor la afişarea partiţiilor de pe disc (partiţie: domeniu definit pe harddisk pentru sistemul de operare şi pentru date). Acest punct al programului foloseşte la verificarea ocupării ac-

tuale a harddisk-ului. Dacă obţineţi un mesaj de genul "partiţie nedefinită", atunci nu mai puteţi face nimic cu FDISK. Terminaţi lucrul cu acest program şi citiţi în continuare.

- Cînd vă este afişat tabelul partiţiilor, verificaţi dacă partiţia "C:" este activă. Acest lucru se întîmplă atunci cînd în coloana **Status** este trecut un "**A**". Dacă nu se întîmplă aşa, alegeţi din meniul principal punctul 2 (activare partiţii). Activaţi partiţia tastînd un "1". Terminaţi programul FDISK şi porniţi din nou calculatorul. Înainte de acest lucru nu uitaţi să introduceţi în unitatea A: o dischetă sistem. Dacă partiţia C: este activă, puteţi accesa harddisk-ul. Salvaţi-vă acum datele şi apoi reformataţi harddisk-ul cu un format de nivel înalt ("High-Level"). Discul va fi apoi din nou gol şi pregătit de lucru.

Doar pentru posesorii de AT-uri: suspendarea SETUP-ului

Posesorii de AT-uri mai au încă o şansă, în comparaţie cu XT-ul el posedă un spaţiu de memorie alimentat printr-o baterie, în care informaţiile rămîn memorate şi după decuplarea calculatorului. Acest domeniu se numeşte "**NVR**" (Non Volatile RAM).

În acest spaţiu de memorie se memorează datele despre configuraţia calculatorului. Dintre informaţiile stocate aici enumerăm: tipul discului instalat în sistem, numărul de capete de scriere/citire şi numărul de unităţi din care este format harddisk-ul. Se poate întîmpla ca informaţiile memorate în NVR să se piardă, fie datorită unei încărcări statice, fie datorită descărcării bateriei de alimentare.

- Presupunînd că informaţiile despre configuraţia calculatorului au fost deteriorate de o încărcare statică se poate întîmpla, de ex. ca sistemul să creadă că dispune de un harddisk de 40 MBytes şi în realitate să dispună doar de unul de 20 MBytes. Un harddisk de 40 MBytes posedă mai multe capete de scriere/citire decît un disc mai mic şi încercarea de iniţializare a harddisk-ului va da greş cel mai tîrziu în momentul în care se va încerca comandarea capetelor de scriere/citire care de fapt nu există. Urmarea: calculatorul intră în grevă.

- Şi mai rău este atunci cînd datorită unei erori în NVR va fi luat în considerare un harddisk de o capacitate mai mică decît cea existentă în realitate. Calculatorul porneşte normal şi se opreşte brusc fără motiv. Singurul remediu în acest caz este modificarea informaţiilor din NVR. Aceasta se poate face cu ajutorul programului SETUP. Acesta poate fi lansat fie de pe o dischetă fie de pe harddisk. El poate fi activat în timpul procesului de iniţializare prin apăsarea unei anumite combinaţii de taste, de obicei: **< Alt T>** , sau **<Del>** sau **<Ctrl PrintScreen>**.

- Dacă AT-ul sesizează combinaţia de taste introdusă, este activată rutina SETUP şi calculatorul va putea fi configurat din nou. Acest lucru se poate face de obicei sub conducerea unui meniu. După ce aţi configurat calculatorul va trebui să opriţi programul. Calculatorul va efectua automat un nou start cu valorile actualizate. Dacă în acest moment calculatorul d-voastră se comportă normal puteţi fi mîndru, aţi reuşit să faceţi o reparaţie excelentă.

Controlul hardware-ului

Dacă n-aţi ajuns la nici un rezultat cu programele SYS, FDISK sau cu SETUP, trebuie să vă obişnuiţi cu ideea că va trebui să desfaceţi calculatorul, pentru a controla conectorii. Dar nu uitaţi că dacă faceţi acest lucru în perioada de garanţie, pierdeţi dreptul la garanţie!

Efectuaţi următoarele operaţii:

- scoateţi ştecherul din priză

- deschideţi calculatorul, de obicei trebuie desfăcute cîteva şuruburi cu cap în cruce.

- localizaţi harddisk-ul. Este uşor de depistat fiind singurul component la care ajung trei cabluri. Două dintre ele sînt tip bandă lată şi sînt conectate unul lîngă celălalt. Cel de-al treilea este cablul de alimentare al harddisk-ului şi are patru fire colorate: unul roşu, unul galben şi două fire negre.

- verificaţi mai întîi dacă firele cablului de alimentare sînt corect introduse în conector

- testaţi ştecherele celor două cabluri tip bandă lată să fie bine fixate la ambele capete (atît la harddisk cît şi la controller)

- mai mult nu puteţi face aici. Asamblaţi calculatorul din nou şi mai încercaţi odată paşii enumeraţi la "primul ajutor".

Cînd nu mai ajută nimic: formatare

Dacă aparţineţi acelei categorii de utilizatori, căreia nici una dintre metodele descrise anterior nu i-a ajutat, atunci vă rămîne ca ultimă şansă doar reformatarea discului, toate datele deţinute fiind pierdute în acest caz. În mod normal un harddisk se formatează o singură dată, în momentul instalării. Această operaţie, mare consumatoare de timp, face harddisk-ul funcţional. Ea serveşte la asigurarea schimbului de date dintre harddisk şi controller pe de o parte, şi pe de altă parte dintre controller şi sistemul de operare.

Pentru ca schimbul de date dintre cele trei unităţi să fie posibil, sînt necesare două formatări ale harddisk-ului: o formatare "Low Level" care asigură comunicaţia dintre controller şi disc şi o formatare "High Level" care asigură accesul sistemului de operare pe harddisk.

Pentru formatarea High Level harddisk-ul trebuie să fie deja formatat Low Level şi la pornire nu are voie să dea mesaje de eroare refe-

ritoare la harddisk. Cum acest lucru nu este valabil, în cazul exemplului nostru, trebuie să începem mai întîi cu o formatare "Low Level". În această privinţă între XT şi AT există o diferenţă esenţială. De aceea vom descrie separat rutinele de formatare pentru cele două echipamente.

Formatarea Low Level la XT-uri

Controller-ul de harddisk la XT-uri este altfel construit decît la AT-uri. Atunci cînd pe piaţă au apărut primele PC-uri, nimeni nu s-a gîndit că într-o bună zi preţurile harddisk-urilor şi a altor memorii va scădea atît de tare încît fiecare calculator va putea fi dotat cu un harddisk. Din acest motiv PC/XT-ul nu are o comandă proprie, "on Board", pentru harddisk (deci cuprinsă în BIOS-ul propriu). Harddisk-ul va fi comandat prin controller, de aceea toată munca de formatare trebuie executată de acesta.
Ce este de făcut:

- porniţi calculatorul avînd în unitatea A: discheta sistem DOS şi aşteptaţi apariţia prompter-ului.
- porniţi programul sistem "Debug". Atunci cînd "Debug" este încărcat şi aşteaptă datele d-voastră, la începutul rîndului va apare semnul "-". Introduceţi următorul rînd (atenţie în acest moment se pierd toate informaţiile de pe harddisk !): "**G=c800:5**". Folosiţi această comandă doar atunci cînd doriţi în mod cert o formatare nouă.
- în continuare apăsaţi pe tasta "**R**". Veţi fi rugaţi să introduceţi diferite valori dependente de controller. Dacă aveţi posibilitatea de a alege o opţiune implicită, faceţi-o, ea uşurează munca unui începător.

- înainte sau după formatare vi se va cere să precizaţi sectoarele defecte (bad tracks). Acestea sînt zone defecte pe disc, apărute încă din fabricaţie. O listă a sectoarelor defecte este lipită de obicei pe harddisk. Introduceţi neapărat aceste sectoare defecte, altfel vor putea apărea erori în date.

- în continuare va trebui să introduceţi factorul "interleave". Explicarea acestei noţiuni nu intră în tema pe care şi-o propune acest articol (pentru informaţii suplimentare consultaţi "if" 1/90), dar atenţie: un factor interleave greşit introdus poate reduce foarte mult viteza de lucru a calculatorului. La un XT ar trebui ales un factor interleave cuprins între 3 şi 5, în funcţie de viteza calculatorului. Cu cît calculatorul este mai lent cu atît factorul interleave trebuie să fie mai mare. Dacă formatarea Low Level a decurs normal, atunci v-aţi atins deja scopul pe jumătate.

Formatarea Low Level la AT-uri

În cazul AT-urilor, rutinele necesare pentru harddisk sînt gata înglobate, de aceea controller-ul harddisk-ului unui AT nu are "inteligenţă" proprie, cum este cazul la XT-uri. În principiu pentru formatarea Low Level a harddisk-ului unui AT este necesar un software special. Unii producători de harddisk-uri oferă acest soft special pentru formatare odată cu produsul. De ex. discurile firmei Seagate sînt livrate împreună cu programul "Disk Manager" al firmei Ontrack. Unele discuri ale producătorului Maxtor sînt livrate împreună cu programul "Speedstore".

Crearea unei partiţii cu FDISK

Dacă formatarea de bază s-a derulat cu succes, porniţi calculatorul cu o dischetă sistem în unitate şi încărcaţi programul FDISK

- alegeţi din meniu punctul 1 (crearea unei partiţii DOS sau a unei unităţi logice)

- în continuare alegeţi din nou punctul 1 (crearea unei partiţii DOS primare)

- răspundeţi afirmativ la întrebarea dacă doriţi să puneţi la dispoziţia partiţiei DOS primare întregul spaţiu de memorie

- FDISK vă va informa că sistemul va trebui pornit din nou. Acţionaţi o tastă şi aşteptaţi pînă cînd calculatorul se va prezenta ca de obicei

Formatarea High Level

Pînă în acest moment a fost efectuată formatarea Low Level pentru ca harddisk-ul şi controller-ul să poată comunica între ele. A urmat crearea unei partiţii, deci a acelui domeniu de disc pe care vor fi memorate datele. Această partiţie trebuie formatată din nou pentru a putea fi folosită de sistemul de operare DOS. Această formatare High Level se face cu ajutorul programului DOS
FORMAT.
Pentru aceasta:

- porniţi programul FORMAT de pe discheta d-voastră sistem, introducînd comanda:
"**FORMAT C: /S**". Parametrul "/S" comunică programului FORMAT faptul că pe harddisk trebuie depus un sistem de operare, pentru ca în viitor calculatorul să poată porni şi fără dischetă sistem. În acest caz FORMAT face harddisk-ul disc sistem.

- dacă formatarea s-a terminat cu succes, puteţi folosi harddisk-ul pentru înmagazinarea datelor şi programelor d-voastră. Ar fi de folos dacă aţi avea o copie de siguranţă a tuturor datelor d-voastră dinainte de a se defecta harddisk-ul, sau de a-l formata. În acest caz aţi putea transfera această copie pe harddisk.

După cum vedeţi există mai multe "trucuri" pentru salvarea informaţiilor de pe un harddisk defect. Nu disperaţi deci atunci cînd apare (din cînd în cînd) cîte o eroare la citire.

(Computer Live 1/91,
Joerg Engelsing)

Schiţă de post:

Administratorul unei reţele locale

Administratorul reţelei poartă răspunderea pentru reţeaua instalată. Fiind vorba de o meserie nouă, încă nu există o descriere exactă a acestui loc de muncă. În S.U.A. însă, această descriere deja s-a făcut.

Deşi "administrator de reţea" este un titlu potrivit pentru cineva care este răspunzător de o reţea locală, acesta este un post "nedefinit". În multe întreprinderi însă, un post de pe statul de funcţiuni poate fi ocupat abia după ce există o descriere exactă a postului. Deseori, prima activitate a noului administrator constă în formularea unei descrieri a propriului loc de muncă.

Descrierea locului de muncă ar trebui să cuprindă sarcinile tehnice şi administrative precum şi conţinutul pregătirii şi devenirea profesională. Prezentăm în continuare un model de astfel de descriere de loc de muncă, aşa cum a fost ea concepută în S.U.A. pentru o întreprindere care tocmai şi-a instalat prima reţea. Descrierea porneşte de la premisa că administratorul de reţea răspunde de selecţia, instalarea şi funcţionarea reţelei locale. Şi personalul de întreţinere (suport) îi este subordonat.

Cerinţe generale

Munca independentă este premisa de bază pntru această poziţie. Deoarece profesiunea este încă tînără, administratorul de reţea nu se poate sprijini pe un 'tezaur' de experienţă existentă. El trebuie să arate siguranţă de sine şi să aibă iniţiativă, pentru a realiza obiectivele cotidiene şi cele pe termen lung. El administrează bugetul reţelei; o secţie de întreţinere (suport) îi este subordonată. Răspunde de planificarea, achiziţionarea, instalarea şi operarea unui LAN nou, de modificări la reţele deja existente şi de racordarea de LAN-uri la reţeaua globală a întreprinderii prin "gateway"-uri şi alte interfeţe.

Administratorul de reţea acţionează ca interfaţă între conducere, utilizatori şi producători atît în ce priveşte problemele tehnice cît şi cele administrative. Pentru a rămîne "la zi", administratorul de reţea trebuie să se instruiască şi să se perfecţioneze continuu.

Răspundere globală

Administratorul de reţea este un manager tehnic, care răspunde şi de formarea unei secţii de întreţinere (suport). Pînă la încadrarea personalului de suport, administratorul preia el serviciile descrise mai jos. După încadrare, răspunderea principală a administratorului constă în planificarea strategică şi supravegherea funcţionării reţelei.

Activităţi administrative

Administratorul de reţea cunoaşte hard-ul instalat în reţea, sistemul de operare, cablarea şi aplicaţiile care se folosesc. Menţine legătura cu producătorii, "VAR"-i (value added reseller-s - revînzători a unor produse cărora le-au sporit valoarea adăugînd ceva de la ei) şi integratori (cei care vînd produse rezultate din integrarea unor subproduse) din zona produselor şi serviciilor LAN.

El concepe specificaţii funcţionale pentru hard-ul de reţea, sistemul de operare, cablări, legături spre un host (calculator "gazdă", de obicei de putere mai mare) şi aplicaţii. Valorifică ofertele comercianţilor şi prezintă conducerii concluziile.

Administratorul de reţea stabileşte bugetul necesar şi justifică selecţia echipamentelor necesare. El alege producătorii şi se informează asupra competenţei tehnice a acestora, stabilităţii lor financiare şi a calificării de suport-întreţinere. Administratorul de reţea comandă hardul şi softul selectat. El planifică şi coordonează instalarea cablului, a hard-ului de reţea şi a sistemului de operare a reţelei.

Activităţi tehnice

Administratorul de reţea planifică şi coordonează întreaga desfăşurare a lucrărilor, din faza testărilor şi pînă la punerea definitivă în funcţiune. El supraveghează activităţile terţilor producători şi furnizorilor.

Administratorul instalează, supraveghează şi modifică organizarea datelor pe server. Modul de organizare a datelor se elaborează în colaborare cu utilizatorii şi conducerea întreprinderii. Folosind 'unelele' specifice pe care le oferă sistemul de operare al reţelei, administratorul instalează caracteristicile utilizatorilor, parolele, directoarele utilizatorilor şi drepturile de acces.

Dacă se dovedeşte necesară stabilirea unor responsabili de reţea proprii pentru diferite grupe de muncă, atunci administratorul de reţea îi alege pe aceştia şi îi pregăteşte. În afară de asta, este disponibil pentru asistenţă tehnică a acestor responsabili de echipă. Administratorul de reţea elaborează proceduri pentru integrarea sau eliminarea unor echipe, proiecte sau utilizatori individuali. Funcţie de necesităţi, administratorul poate delega diferite activităţi înspre responsabilii de echipă şi supraveghează executarea acestora.

Administratorul de reţea planifică transferul de date din alte sis-

teme în reţeaua locală. El coordonează instalarea interfeţelor de reţea şi eventualele legături spre un host.

Administratorul de reţea este răspunzător de instalarea hardului. El contractează eventual servicii de întreţinere şi supravegează executarea lucrărilor de întreţinere. Dacă este necesar, înlocuieşte cartele de interfaţă pentu reţea şi conectori. Capacităţile libere în sistem şi implementarea unor facilităţi care să conducă la creşterea performanţelor sînt supravegheate de el.

Administratorul este răspunzător de înmagazionarea şi depozitarea echipamentelor şi materialelor auxiliare necesare (hîrtie, toner, mine pentru plotter etc.)

Administratorul de reţea este răspunzător de funcţionarea cotidiană a reţelei. El elaborează metode pentru recunoaşterea şi corectarea din timp a unor malfuncţii. Poate delega aceste răspunderi asupra personalului de suport. Elaborează o bază de date de configurare.

Administratorul de reţea supraveghează prestaţiile serverului cu ajutorul jurnalelor de bord şi a unor analize de tendinţe. Supraveghează toate modificările la server, la cablare şi la alte resurse ale reţelei. El controlează resursele libere şi asigură existenţa unor capacităţi suficiente pentru aplicaţii şi fişiere noi.

Supravegherea funcţionării reţelei

Administratorul de reţea planifică şi instalează aplicaţii noi pe server şi îi instuieşte pe responsabilii de echipă şi utilizatorii finali în utilizarea acestor programe. El testează programe noi, pentru a fi sigur că acestea nu încarcă prea mult reţeaua sau că nu intră în conflict cu alte programe. El elaborează utilitare soft pentru administrarea reţelei, cum ar fi utilitare de imprimare sau fonturi precum şi fişiere de comenzi.

Strategia întreprinderii, administratorul stabileşte procedurile de asigurare a securităţii datelor. El planifică şi supraveghează salvarea regulată a datelor (backup). Păstrează protocoale ale acestor activităţi şi verifică corectitudinea benzilor pe care se fac salvările. Date deja şterse vor fi restaurate la solicitare de pe copiile de siguranţă de către administrator.

Pregătire şi suport

Administratorul de reţea elaborează metode şi proceduri pentru administrarea reţelei, ţinînd seama de strategia întreprinderii. Planifică şi elaborează proceduri pentru instruire relativ la reţea. Utilizatorii finali sînt instruiţi în permanenţă. În acest scop, administratorul elaborează, înnoieşte şi distribuie documente relativ la modificările în reţea.

Administratorul de reţea angajează colaboratori pentru administrarea reţelei şi îi instruieşte pe aceştia. El elaborează un sistem de asistenţă tehnică, care este la dispoziţia utilizatorilor pentru întrebări.

Dacă mai mulţi utilizatori se confruntă cu aceeaşi problemă, administratorul stabileşte dacă este vorba de o funcţionare greşită a reţelei sau dacă utilizatorii nu ştiu să folosească corect aplicaţia. Corectează problema în reţea sau respectiv îi instruieşte pe utilizatori relativ la folosirea softului.

La modificări ale reţelei, administratorul de reţea se îngrijeşte ca acestea să se facă cu deranjul minim al utilizatorilor. El elaborează un sistem de gestionare a problemelor, pentru a recuoaşte timpuriu eventuale malfuncţii şi pentru a le putea evita. Pentru a avea la dispoziţie datele necesare unei recunoaşteri timpurii a erorilor, administratorul elaborează un sistem de control a configuraţiei.

Experienţa profesională şi pregătirea

Pentu această activitate, o premisă este absolvirea unui institut superior informatic. Administratorul trebuie să posede forme de exprimare foarte bune, atît în scris cît şi oral.

Se presupune că administratorul de reţea are o experienţă de cel puţin doi ani şi cunoaşte o mare varietate de calculatoare personale. În activitatea anterioară, ar trebui să fi avut deja responsabilităţi în ceea ce priveşte hard-ul şi softul, suportul unor aplicaţii precum şi instruirea utilizatorilor. Administratorul trebuie să posede cunoştinţe profunde despre sistemele de operare folosite în întreprindere. Sînt necesare deasemenea cunoştinţe relative la comunicaţii cu un host şi protocoale de comunicaţie.

Administratorul de reţea ar trebui să aibă 1 pînă la 3 ani de experienţă în ce priveşte responsabilităţi relative la personal şi buget.

Administratorul de reţea este un manager.

Grosul activităţii sale se află în domeniul administrativ. De aceea, cunoştinţe tehnice cuprinzătoare sînt o premisă. Hotărîtoare însă este capacitatea organizatorică şi motivaţională a administratorului.

De aceea, cînd se elaborează o astfel de descriere de loc de muncă, este important să se sublinieze în mod deosebit activităţile administrative. Căci activităţile tehnice ale administratorului nu sînt totdeauna pe deplin inteligibile pentru conducerea întreprinderii (management). Cheia succesului administratorului rezidă deci în a putea fi considerat de către conducerea întreprinderii ca "unul de-ai noştri", şi nu ca un tehnician "pur".

(LAN Magazin 2/90,
Ira S. Hertzoff)

Ira Hertzoff este preşedinta firmei Tholian Holdings Inc., întreprindere de management şi consultanţă pentru reţele cu sediul în Columbus, Ohio, S.U.A.

A luat fiinţă

Institutul European contra Viruşilor

Experţi în viruşi ai calculatoarelor din întreaga Europă s-au întîlnit pentru prima dată la congresul Percomp în Hamburg, pentru a se sfătui relativ la măsurile ce ar trebui luate pentru combaterea temuţilor viruşi. Nume atît de renumite printre iniţiaţi ca cel al prof. dr. Klaus Brunstein, Universitatea Hamburg sau al specialistului islandez Fridrik Skulason, Universitatea Reykjavik s-au numărat printre participanţi alături de expertul particular în securitatea şi regăsirea datelor dr. Alan Solomon, Marea Britanie. Şi producători de soft de renume ca Digital Research şi Borland au fost prezenţi. Reprezentanţi ai poliţiilor germane şi olandeze au remarcat că problematica viruşilor a rămas (încă) ascunsă. Utilizatorii afectaţi sînt în mod evident mai interesaţi de limitarea daunelor decît de aflarea vinovaţilor.

Ca probleme centrale s-au discutat nomenclatura la apariţii multiple a unor viruşi identici şi rezultînd din aceasta, identificarea "pasagerilor clandestini". Cu mare interes a fost primită intervenţia lui Skulason, care a apreciat că numărul diverşilor viruşi de calculatoare şi al variantelor modificate va ajunge în viitorul apropiat la 1000. Calculul său relativ la perioada de înjumătăţire în care devin cunoscuţi noi viruşi este de circa 10 luni. În decembrie 1990 erau cunoscuţi circa 250 de viruşi. Pentru producătorii de software antivirus acest lucru înseamnă că schimbul de viruşi detectaţi, experienţa în analiza acestora precum şi frecvenţa actualizărilor sînt supuse unor noi dimensiuni.

O altă dificultate o ridică apariţia unor viruşi de tip nou, care se autocriptează. Analiza unui astfel de cod poate necesita un atare efort de muncă încît nu mai rămîne timp pentru combaterea unor viruşi noi. Ca o altă circumstanţă agravantă se adaugă faptul că între timp programarea de viruşi pare să fi devenit un fel de sport de masă. După cum a relatat Veselin Boncev, de la Academia de Ştiinţe Sofia, în Bulgaria există o cutie poştală care serveşte exclusiv schimbului de viruşi de calculatoare. Nenorocirea cu această cutie poştală (mailbox) este că numai acela are voie să-şi copieze viruşi pe calculatorul propriu care a transferat în cutia poştală un virus necunoscut pînă în momentul respectiv. Acest lucru nu poate fi înţeles decît ca o provocare de a programa singur viruşi noi.

Necesitatea unei diviziuni a muncii pe plan internaţional a fost general acceptată. Urmînd propunerilor făcute de dr. Solomon, a fost înfiinţat EICVR: European Institute for Computer Virus Research. Această întreprindere operînd pe baze private va permite pe viitor cooperarea forurilor private, oficiale şi universitare în combaterea viruşilor. În afară de înfiinţarea unei bănci de viruşi (care va fi însă numai la dispoziţia unor experţi antivirus recunoscuţi) va fi reglementată şi denumirea viruşilor noi apăruţi. Ţinînd seama de internaţionalitatea prelucrării electronice a datelor, este avută în vedere înfiinţarea unui cartel antivirus mondial.

În cercurile de specialişti, geneza EICVR e privită ca o senzaţie. Darea mai departe a unui cod de virus, în cunoştinţă de cauză, fie chiar şi numai unui cerc restrîns de primitori avizaţi, era considerată pînă acum ca fiind un delict. Creare unui for care să supravegheze deschis un astfel de schimb ar putea avea ca efect o încredere mult mai mare în softul de protecţie folosit în fiecare caz - oricare ar fi producătorul lui. Din punctul de vedere al utilizatorilor, înfiinţarea EICVR reprezintă o iniţiativă salutară.

(PC Magazin, 3/91)

Iată, din nou, cîteva preţuri orientative (ianuarie 1991):

Sisteme	Preţ de stradă (no-name) (mărci)	Aparate de marcă, ca de ex.
AT 80286 -CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy	1498	IBM-PS/2-50 cca 5800-7200
386SX 80386SX-CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy, VGA-color	2066	Compaq386S cca. 7700-8700
386/33 80386-CPU, 2 Mbyte RAM, harddisk 150 Mbyte, o unitate floppy, VGA color	4734	Compaq 386/33 cca. 17.000-22.000
486/25 80486-CPU, 4 Mbyte RAM, harddisk de 320 Mbyte, o unitate floppy, VGA color	9834	HP Vectra 486 cca 31.000-34.000

Poşta redacţiei

"Deşi nu am acces (decît sporadic) la un PC (profesional), revista Dvs. mă interesează pentru că vine în întîmpinarea intenţiilor mele de viitor: este actuală, bine scrisă şi ... unică în ceea ce priveşte cantitatea informaţiilor de calitate!

Mica publicitate

Vînd AT 286 tact 12 MHz, driver 5,25" de 1,2 Mbyte, harddisk 40 Mbyte, 28 ms, 1 Mbyte RAM, Hercules, monitor monocrom, mouse - totul la preţul de 2500 DM. Tel. 961/48631 - Timişoara.

Cumpăr imprimantă Robotron K6313 sau K6414 şi unitate de disc pentru Commodore 64. tel. 807059 sau C.P. 69-130, Bucureşti.

G&Condor oferă: calculatoare personale, periferice, programe aplicaţii, afişaje reclamă dinamică, lucrări de instalare, asistenţă tehnică, service, elaborare proiecte, standarde, documentaţie tehnică pentru echipamente electronice (editare color) Tel. 976/21968

Specialist software : gestiune de date, interfeţe utilizator, grafică. Lucrez TurboPascal (fişiere în acces direct, fişiere index, gestiune heap), limbaj de asamblare. Mari disponibilităţi de continuă perfecţionare (26 ani). Ofer serviciile unei firme străine interesate. Tel. 940/41177

Firma "Hello" vă oferă proiectare asistată de calculator, contabilitatea întreprinderilor, instruire diferite sisteme de calcul. Tel. 90/77.05.86 (Bucureşti)

Execut programe C, Pascal, dBase. Aştept oferte serioase. Tel. 90/654186

Doresc să colaborez cu posesori de **Commodore 64**. Paraschiv V. Teodor, Cezar 63, 6200 Galaţi

De ce am considerat necesar să vă expediez aceste cîteva rînduri ?! Din cauza ..."profesiunii de credinţă" din nr. 1, care îi condamnă pe posesorii de HC-uri la aşteptare (...) majoritatea acestora nu au nici cea mai mică idee despre modul cum l-ar putea folosi (eficient).

Sînt convins că nu aţi avea decît de cîştigat (enorm!...) acordîndu-le (totuşi ...) un minim de spaţiu (= atenţie); pentru că, mă tem, nimeni nu se va încumeta să "scoată" o revistă numai pentru HC-uri, iar ca să vă concureze - tratînd şi PC-uri - ar fi (cred) sinucidere curată !

Ce vor HC-iştii?! Ce le trebuie lor ?! (Ce este HC-iştii şi ce vrea ei !)

TOTUL !

Adică: noţiuni elementare (teoria informaţiei, algoritmi, limbaje; sisteme de operare - mai puţin); domenii de utilizare (cu multe sugestii practice), tendinţe etc.

Să nu uităm că posesorii de HC-uri sînt viitorii (siguri) utilizatori de PC-uri (deci - potenţiali clienţi ai firmei Dvs, fie şi numai prin revista "if"!).

Pentru că majoritatea utilizatorilor de PC-uri din România le pot folosi numai la serviciu (...), n-ar fi lipsit de interes (pentru ei) să discutaţi în revistă şi probleme specifice anumitor domenii - cu care ei se întîlnesc în cadrul sarcinilor de serviciu; chiar dacă prin aceasta aţi face concurenţă revistelor "emanate" de "emanaţiile" I.C.I. Concurenţa aceasta nu v-ar dezavantaja - cel puţin judecînd după conţinutul de pînă acum al revistei Dvs. !

Precizez, pentru clarificarea lucrurilor, că nu posed HC, nici PC şi nici nu am acces la PC-uri (decît accidental); iar această situaţie va mai dura pînă ce-mi voi putea permite să investesc într-un PC, acţiune în care voi fi ajutat (enorm) de tabelele publicate de Dvs.(...)

Cu îngăduinţa Dvs., mi-aş permite să sugerez cîteva idei care vă vor spori, cred, audienţa:

1. continuarea orientării spre "compatibile IBM", avînd în vedere că această firmă pătrunde masiv în întreprinderi (MS-DOS)

2. atenţie specială Bucureştiului - o piaţă imensă (even if ...)

3. un ciclu despre arhitectura sistemelor de operare (util mai ales pentru elevii de liceu din oraşele "provinciale"

4. un ciclu despre limbaje de asamblare (bazat pe cel mai răspîndit)

Ambele cicluri (în serial) una, două pagini/număr; cursul de "macro" - împănat cu exemple şi probleme propuse.

Succes!

Cu stimă,"

Coveanu Ion, Slatina

"Ceea ce îmi place cel mai mult la revista editată de dumneavoastră este rubrica cititorului, pe care n-am mai întîlnit-o pînă acum la nici o altă revistă ."(...)

Manea Alexandru, Dorohoi

"Ideea voastră de a specializa revistele mi s-a părut excelentă. Să vedem cine va lansa o revistă pentru home-computere, care se pare că au o mare răspîndire la noi.

Vă propun iniţierea unor concursuri pentru utilizatori. Pentru cîteva premii simbolice, veţi putea avea surprize plăcute şi o colaborare mai strînsă cu cititorii.(...)"

Cristian Popovici, Bucureşti

Nota redacţiei: dorim lansarea unor concursuri pentru cititori, dar încă nu am găsit o modalitate care să ni se pară tentantă. Desigur, materialele care le publicăm sînt plătite (e drept că mai degrabă simbolic). Pînă ce punem ceva pe roate, iată totuşi o idee pentru cititorii noştri: la rubrica "Practică", similară ca idee cu cea din revista noastră (sau mai degrabă invers), revista PCplusTechnik oferă un premiu de cel puţin 50 DM pentru fiecare "şmecherie" publicată.. Adresa completă este Markt &Technik Verlag AG, Redaktion PC Technik, Stichwort : Tips & Tools, Hans Pinselstrasse 2, 8013 Haar, Germania. Desigur, textele pot fi trimise numai în limba germană sau engleză Noi putem promite doar că îi vom ajuta prin traducere şi reexpediere (acesta să fie "premiul" nostru) pe cei care ne trimit materiale în acest sens, materiale care să ni se pară că într-adevăr aduc ceva interesant şi pe care le vom publica . Pentru materialele nepublicate nu ne putem asuma nici un fel de răspundere.

Stimaţi cititori,

Ca urmare a noilor etape de "liberalizare" a preţurilor, sîntem nevoiţi să scumpim preţul revistei. Astfel începînd cu numărul 3/91, un exemplar va costa 57 lei, inclusiv cheltuielile de expediţie.

Începînd cu numărul 4/91 sîntem nevoiţi să renunţăm la expedierea revistelor cu plata prin ramburs, deoarece este sensibil mai scumpă, mai complicată şi mai nesigură (retururile fiind foarte scumpe). Începînd cu numărul 4/91 revista poate fi procurată numai prin abonament sau de la difuzorii noştri.

Puteţi încheia un abonament la revistă completînd talonul alăturat şi expediind în contul nostru, prin mandat poştal, o sumă calculabilă cu formula:

57 lei * nr. exemplare * nr. de apariţii pentru care doriţi abonament.

Vă rugăm să vă completaţi citeţ adresa şi să daţi, la rubrica "Loc pentru corespondenţă" de pe mandatul poştal, o scurtă explicaţie a sumei expediate (nr. exemplare, nr. apariţii).

Abonamentele intră în vigoare de la data intrării banilor în cont.

Mărturisim că am fi putut încerca să compensăm "liberalizarea" printr-o reducere a numărului de pagini. Volumul informaţiilor de comunicat fiind însă atît de mare încît chiar şi numărul de pagini actual este insuficient am considerat măsura mascării scumpirii printr-o reducere a numărului de pagini ca fiind de neacceptat. (Am fi putut păstra preţul de 39 lei la cca. 20 pagini !). Ţinînd cont şi de lipsa acută de documentaţii, în numărul următor 4/91, vom adăuga încă 4 pagini conţinînd un **memento DR-DOS 5.0.** În funcţie şi de modul în care vor fi primite aceste îmbunătăţiri de către D-voastră, acesta ar putea fi începutul unei noi serii.

Sperăm să rămîneţi în continuare un cititor al revistei "**if**".

Redacţia

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if,** pe o perioadă de |_____| apariţii, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 57 lei. Preţul unui abonament se va calcula cu formula: 57 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

if 3 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

LOGIC

Societate comercială româno-americană

SIBIU, Calea Dumbrăvii 21,tel.(924)46652,45475 Fax:(924)46704
BUCUREŞTI, Bulevardul Tineretului 8-10,tel.(90)754900 Fax:(90)757135
Cont valută BRCE Buc.: 47.38.112.300-2
Cont lei BRCE Buc.: 30.08.112-9
Cont lei BC Sibiu: 30.80.4.03.01

LOGIC vă oferă produsele *BORLAND* în România

* PARADOX 3.5
* PARADOX SQL LINK
* PARADOX RUNTIME 3.5
* PARADOX ENGINE
* QUATTRO PRO 2.0
* QUATTRO PRO SHOW
* TURBO PASCAL PROFESSIONAL 6.XX
* TURBO PASCAL RUNTIME LIBRARY 6.XX
* TURBO PASCAL 6.XX
* TURBO C++ PROFESSIONAL
* TURBO C++ RUNTIME LIBRARY
* TURBO C++
* TURBO C 2.XX
* SIDEKICK 2.0
* TURBO DEBUGGER & TOOLS 2.XX
* SPRINT 1.XX
* SUPERKEY 1.XX

Firma **LOGIC** asigură inregistrarea licenţelor de utilizare si acordă reduceri de preţ pentru versiunile noi.

Important !

Revista "if" îşi propune editarea unui catalog al întreprinderilor (de stat sau particulare) a căror sferă de preocupări are tangenţă cu informatica (producători hard, producători soft, depanatori, distribuitori, comercianţi, etc.). Catalogul va conţine adresele întreprinderilor, o scurtă descriere a obiectului de activitate al acestora şi o prezentare succintă a produselor şi serviciilor oferite. Catalogul va fi distribuit prin reţeaua de difuzare a revistei "if", putînd ajunge astfel direct la persoanele interesate. Întreprinderile interesate de publicarea acestui catalog sînt rugate să ia legătura cu redacţia.

Important !

Căutăm difuzori/distribuitori autorizaţi pentru difuzarea revistei "if".
Condiţii avantajoase ! Rugăm persoanele interesate să ia legătura cu redacţia.

LEUTRON VISION AG, ZÜRICH, ELVEŢIA & MICRO ATCI, TÎRGU MUREŞ,

... partenerii Dvs. în procesări de Imagini - DIP (Digital Image Processing)

După 10 ani de experienţă, spectrul de produse al firmei LEUTRON VISION AG cuprinde :

— **DIP PENTRU APLICAŢII MEDICALE**
Memorii video (Video-Framestore) şi sisteme de procesare de imagini complete, îndeosebi pentru sisteme röntgen C-ARM: MICROSTORE şi VIP100XB pentru norma video HDTV normală precum şi pentru cea de mare rezoluţie, rezolvă total problema suportului röntgen prin captarea instantanee a imaginilor pe monitor, cu sau fără posibilităţi de îmbunătăţire/manipulare a acesteia precum şi arhivare pe harddisk.

— **DIP PENTRU APLICAŢII INDUSTRIALE**
Memorii video (Framestore)/ interfeţe de procesare de imagini pentru sistemele IBM PC/AT sau compatibile; SOFTWARE standard sau la cererea Dvs. ca şi sisteme de procesare de imagini industriale pentru controlul calităţii sau sisteme de viziune pentru roboţi. Interfeţe de procesare ultra-rapide prevăzute cu TRANSPUTER-ele din seria INMOS-IMS T80.

— **DIP PENTRU SISTEMELE VME BUS BAZATE PE MICROPROCESORUL MOTOROLA 68000**
Memorii video/interfeţe pentru VME BUS, SOFTWARE la cererea Dvs. precum şi sisteme complete (turn-key systems).

Creată în 1979, **LEUTRON VISION AG** dezvoltă, produce şi vinde HARDWARE şi SOFTWARE atît ca produse independente cît şi ca SISTEME LA CHEIE. KNOW-HOW-ul complet, începînd cu analiza sistemului şi pînă la nivelul proiectării HARD şi SOFT este realizat de producător, fapt care ne permite să vă satisfacem cerinţele Dvs. specifice, inclusiv pentru produse în serie mică.

Produsele LEUTRON VISION AG funcţionează cu succes în lumea întreagă.

DISTRIBUITOR OFICIAL PENTRU ROMÂNIA

MICRO ATCI
Str. Bartok Bela Nr. 3
CP64, OP 1
RO - 4300 TÎRGU MUREŞ
Tel. : 954 / 20521

dipl. ec. IUDEAN VIOREL
dipl. ing. KALLO TIBOR

CONTACTAŢI-NE ACUM ! NOI REZOLVĂM PROBLEMELE DVS. !

if

CALCULATOARE PERSONALE

4 / 91

(7)

57 Lei

Revistă lunară editată de Micro ATCI Tîrgu Mureş

Complet:
o bibliotecă pentru mouse

Fundamente:
Prelucrarea digitală
a imaginilor

8 Reţele

Seriale:

OOP şi reţele neuronale

Anchetă:

Noutăţi pe piaţa românească

Surpriză: Memento DR DOS 5.0 detaşabil !

AbMod

Ábráhám & Moldovan s.n.c.

vă oferă **tehnologie americană**
inteligenţă europeană
preţuri orientale **la**

produse din domeniile:

- **tehnică de calcul (hardware şi software)**
- **telematică şi birotică**

servicii în aceleaşi domenii:

- **lucrări de cercetare / proiectare**
- **consulting şi asistenţă tehnică**
- **instalare, şcolarizare, service**
- **sisteme "la cheie"**

condiţii de livrare: 2 - 4 săptămîni

Cîteva exemple din lista noastră de preţuri:

XT

<< Preţ sistem complet **89.000 lei >>**
<< cu disk winchester 20 Mb **149.000 lei >>**

AT-286 & AT-386

<<Preţ sistem complet [lei]

AT	Winchester capacitate formatată	monitor + interfaţă		
		mono 14"	EGA 14"	VGA 14"
286	40 Mb	199.000	249.000	299.000
	80 Mb	275.000	325.000	375.000
386	80 Mb	399.000		499.000
	160 Mb	549.000		649.000

Imprimante matriciale

Tipul	Lăţimea	Nr. ace	Preţ [lei]
LC 20	80 car	9 pin	49.000
LC 15	132 car	9 pin	79.000

Mouse serial compatibil Microsoft . . . . . 7.700 lei
Copiatoare FC-2 (Canon) . . . . . 199.000 lei
MS-DOS v4.01 . . . . . 16.000 lei

Cei interesaţi pot obţine gratuit lista de preţuri adresîndu-se la

AbMod Sediul central:

str. Minerilor nr. 42, 3400 Cluj-Napoca, tel. 951-56607

AbMod Sucursala Oradea:

str. Decebal nr. 7, bl. D8, ap. 11, 3700 Oradea, tel. 991-60278

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

La realizarea acestui număr au colaborat:

ing. Attila Darvas,
ing. Iosif Fettich,
ing. Alin Flaidăr
ing. Tibor Kallo
ing. Ingrid Maier,
ing. Romulus Maier,
ing. Emil Palade
Colaboratori externi:
ing. fiz. Luminiţa Todor

Tiparul: tipografia Tîrgu Mureş
Tipografi: Halaţiu Mihai,
Cormoş Mircea

Revista apare lunar.

Preţ: 57 lei

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/17024, fax 954/35208,
telex 65354.

Manuscrise originale sau listing-uri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Facerea

1. La început patronul a făcut firma. Şi pentru că trebuia să poarte un nume, i-a spus Micro ATCI.

2. "Pămîntul era pustiu şi gol; peste faţa adîncului de ape era întunerec" şi pasionaţii ştiinţei calculatoarelor cătau cu jind la lumina aducătoare de înţelepciune a altor meleaguri.

3. Şi ne-am zis: "Să facem o revistă dedicată calculatoarelor personale!" Şi ne-am apucat s-o facem.

4. Patronul a văzut că ideea era bună; şi patronul a despărţit întreprinderea în două.

5. O parte şi-a păstrat şi pe mai departe numele şi se ocupă în continuare cu comercializarea, instalarea, depanarea calculatoarelor şi cu tot ce-i mai trebuie unui posesor de calculator personal ca să se simtă bine. O altă parte s-a numit "editare revistă" şi pentru că şi aceasta trebuia să poarte un nume, cu sprijinul naşului - Sorin Bordeianu - a fost botezată "revista **if**".

6. Şi cum din revistă nu puteau lipsi cîţiva stropi din lumina "putredului (de bogat) Occident", şi nepotrivindu-ni-se rolul de guru, şi pentru că editura Markt&Technik ne-a înţeles oful, am încercat o (nu foarte originală) formulă de apariţie.

7. Astfel au fost mai multe seri, şi apoi într-o bună dimineaţă a apărut numărul unu.

8. Şi cititorii au început să ne scrie şi am văzut că este bine. Că este bine şi că mai încape!

9. Şi după alte (multe, prea multe) seri (şi nopţi) într-alte bune dimineţi au apărut numerele doi, trei, patru, cinci şi şase.

10. Şi cum încă nu ne-am sfîrşit lucrarea, al şaptelea număr n-a fost de "nemuncă". Cu ce încercăm să vă captăm atenţia ? Vă recomandăm:

- - cîteva reţele ieftine
- - biblioteca de mouse
- - memento (detaşabil) DR-DOS 5.0
- - ancheta relativă la ce şi de unde se poate cumpăra.

11. Sperăm că veţi rămîne alături de noi, căci: "Nu este bine ca" informaticianul "să fie singur"; el are nevoie de "un ajutor potrivit pentru el". Iar noi pe cine-am mai putea ispiti să se înfrupte din "pomul cunoştinţei binelui şi răului" ? Cui am mai putea spune: "Sînt măr de lîngă drum, şi fără gard..." ? Concluzia: Only "together we are strong !"

12. Sperăm, deci, că vor urma şi alte numere. Acesta ar fi un prim bilanţ. E mult ? E puţin ? Timp este să faci un lucru bine, dar timp este şi să greşeşti. Timp este să ţi se arate greşeala (cînd n-o vezi şi singur), timp este s-o recunoşti şi timp este să te îndrepţi. Vom încerca în această lume nebună, nebună, ne-bună, să vă ajutăm să dobîndiţi un strop de mulţumire măcar pe tărîm profesional.
Căci dacă informatică nu e, nimic (în economie) nu e !

ing. Romulus Maier

Cuprins

Ştiri

- Produsele anului 1990 . . . . . pag.4
- Harddisk cu acceleraţie pag.6
- Alternative pentru reţea pag.6
- Advance Technology Award 1990 . pag.6
- Conectare fără cablu pag.6
- Compaq învingător la Las Vegas . . pag.6
- OS/2-ul a murit - trăiască OS/2! . . . pag.7
- i856 şi i846sx pag.7
- Orice AT 286 poate deveni un 386SX pag.7
- Computer Panoráma pag.7

Magazin

Prietenul meu calculatorul pag.8

"Calculatoarele sînt proaste". Această sentinţă a grupului rock "Spliff" din Berlin are deja cîţiva ani vechime, dar este încă actuală.

IASTED
- organizaţie ştiinţifică internaţională accesibilă şi specialiştilor români . . . pag.10

Ce (şi de unde) putem cumpăra pag.11

Hardware

Sînt SX-urile mai bune decît AT-urile ? pag.14

Salvarea datelor pe bandă magnetică
- cum lucrează un streamer pag.15

Reţele

Reţele de PC-uri la preţuri avantajoase pag.18

- 1.Desklink - Kirschbaum Link pag.20
- 2. Lantastic Z pag.20
- 3. Reţea Kirschbaum pag.21
- 4. Fischernet Starterkit pag.22
- 5. Lantastic pag.22
- 6. Invisible Ethernet pag.23
- 7. Novell ELS I şi ELS II pag.23
- 8. Reţea XIRCOM pag.24

Noţiuni de bază

Bazele prelucrării de imagini pe calculatoare personale pag.25

Software

Formate grafice pag.30

De această dată vă vom prezenta modul în care puteţi transporta grafice între diferite programe şi sisteme de calcul.

- Hijaak pag.30
- XPort pag.30
- Bridge pag.30
- VPIC 2.8 . . . . . pag.30
- GCP pag.31
- Micrografx Designer 3.01 . . . . . pag.31
- Corel Draw 1.2 . . . . . pag.31
- Art & Letters 3.0 . . . . . pag.31
- Windows Clipboard pag.32
- Răsplata ostenelii pag.32

Grafică: de la un sistem la altul pag.32

- PC - Macintosh pag.32
- PC - Amiga pag.32
- PC - ST pag.32
- Alte legături pag.32

Designer 3.01 . . . . . pag.33

Corel Draw 2.0 . . . . . pag.34

Cursuri

OOP - Partea a doua pag.36

Reţele neuronale -Partea a II-a . . . pag.42

Practica

Elemente de programare a mouse-ului pag.46

dBase cu mouse pag.51

Rubrici

- Caseta redacţiei
- Editorial
- Jargon - Vocabularul esenţial . . pag.52
- Din scrisorile sosite la redacţie . pag.54
- Mica publicitate pag.54

Produsele anului 1990

La începutul acestui an, revista **Computer Persönlich** a lansat o anchetă de amploare, dotată cu multe şi substanţiale premii, pentru a determina produsele hard şi soft ale anului 1990, conform aprecierilor cititorilor revistei. 15356 de cititori şi-au făcut timp pentru completarea formularelor. Vă prezentăm şi noi "laureaţii", pe categorii:

Hard

80486 PC
1. Dakota 486/33 EISA
2. Hewlett-Packard Vectra 486
3. Tandon 486/25

80386 PC
1. Dell System 333D
2. Compaq Deskpro 386/33e
3. Siemens-Nixdorf 8810/80

80386SX PC
1. Compaq Deskpro 386s
2. Kyocera Multilight IIIsx
3. Amstrad PC 3386SX

80286 PC
1. Hewlett-Packard Vectra 286/12
2. Schneider Euro AT
3. Vobis Highscreen 286

Laptop (conectat la reţea)
1. Toshiba T5200
2. IBM P70 386
3. Sharp PC-8041

Laptop (independent de reţea)
1. Compaq SLT 386s/20
2. Toshiba T3100SX
3. Amstrad ALT-386SX

Notebook
1. Sharp PC-6220
2. Toshiba T1000XE
3. Compaq LTE/286

Pocket Organizer
1. Sharp IQ-7100M
2. Atari Portfolio
3. Psion Organizer

Cartele grafice VGA
1. Sigma VGA Legend
2. Video Seven 1024i
3. Video Seven VRAM

Cartele grafice de rezoluţie mare
1. Hercules Graphics Station Card
2. Spea Gallery
3. Spea FGA

Hard de prelucrare imagini
1. Digithurst Microeye
2. Fast Screen Machine
3. Truevision Targa

Monitoare
1. NEC Multisync 3D
2. EIZO Flexscan 9070S
3. NEC Multisync 4D

Ecrane mari
1. NEC Multisync 5D
2. Sony GDM-1963
3. EIZO Flexscan 5500

Imprimante matriciale
1. NEC Pinwriter P60
2. Star LC 24-200
3. Epson LQ-850

Imprimante laser
1. Hewlett-Packard Laserjet III
2. NEC Silentwriter 2 S60P
3. Hewlett-Packard Laserjet IIP

Imprimante cu jet de cerneală
1. Hewlett-Packard Deskjet 500
2. Epson SQ-2550
3. Canon BJ-130

Mouse-uri
1. Microsoft Mouse
2. Logi Pilot
3. Genius GM-6000

Scannere alb-negru
1. Logi Scanman Plus
2. Hewlett-Packard Scanjet
3. Marstek 800

Scannere color
1. Epson GT 4000
2. Microtek MSF 300Z
3. Sharp JX-400

Plottere
1. Hewlett-Packard 7550
2. Roland DXY 1200
3. Sekonic SPL 800

Modemuri şi cuploare acustice
1. Elsa Microlink 410T2
2. Dr. Neuhaus Fury 2400
3. Hayes Smart Modem

Produsul hard cel mai inovator
1. Next NeXTstation
2. Intel i486
3. Sun Sparcstation

Soft

Sisteme de operare
1. MS-DOS 4.01
2. DR-DOS 5.0
3. OS/2 1.2

Suprafeţe utilizator grafice
1. Microsoft Windows 3.0
2. Geoworks Ensemble
3. Motif

Sisteme de operare în reţea
1. Novell Netware
2. OS/2-LAN-Manager
3. LANtastic

Programe de prelucrare texte
1. Word 5.0
2. Word for Windows 1.1
3. Word Perfect 5.1

Programe de calcul tabelar
1. Microsoft Excel 2.1
2. Borland Quattro Pro 2.0
3. Lotus 1-2-3 3.1

Baze de date
1. dBase IV
2. Borland Paradox 3.0
3. Clipper 5.0

Programe de administrare proiecte
1. Project for Windows
2. Timeline
3. Harvard Project

Programe de prezentare grafice
1. Harvard Graphics
2. Microsoft Powerpoint
3. Micrografx Charisma

Programe de grafică şi design
1. Corel Draw
2. Micrografx Designer
3. GEM-Artline

Programe de pictură şi desen
1. PC Paintbrush
2. DeLuxe Paint II
3. Windows Paintbrush

DTP
1. Aldus Pagemaker
2. Ventura Publisher
3. Timeworks

CAD
1. AutoCAD
2. MegaCAD
3. Autosketch

Programe de recunoaştere texte
1. Caere Omnipage
2. Logi Catchword
3. Recognita

Programe de regăsire texte
1. AskSam
2. Lars
3. Vq

Limbaje de programare
1. Turbo Pascal 5.5
2. Turbo C++
3. Microsoft C 6.0 PDS

Utilitare
1. PC Tools 6.0
2. Norton Utilities
3. Norton Commander

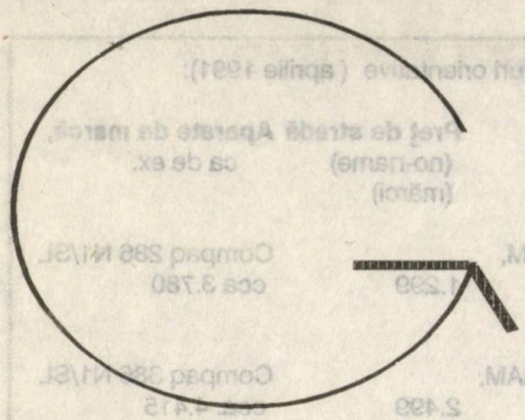
Programe antivirus
1. Virus Scan
2. AntiVir
3. Virusscan

Programe de comunicaţie
1. Procomm Plus
2. Telemate
3. Exis-Telix

Programe de preluare date rezultate din măsurători
1. Asyst
2. Notebook
3. Signalys

Pachete integrate
1. Works 2.0
2. Open Access
3. Framework

Softul cel mai inovator
1. Windows 3.0
2. Toolbook
3. Turbo Pascal 6.0



GENESYS

SZINTEZIS COMPUTER GYÖR UNGARIA

GENESYS ORADEA ROMANIA

str. Republicii nr. 35
3700 Oradea
tel. 991/31516

Denumire	(N)	Preţ (Valută)	Preţ (Lei)
Calculatoare XT			
GF XTT,,640k,,Mo	$	311	78486.3
GF XT,360k,640k,20M,Mo	$	693	174890.7
4D XTTbill,,640k,,,Arcn.	$	600	151420.5
Calculatoare AT 286			
GF AT-12,,1M,,Mo	$	366	92366.5
GF AT-12,1.2M,1M,40M,Mo	$	938	236720.7
GF AT-12,1.2M,1M,80M,Mo	$	1366	344734.0
GF AT-16,,1M,,Mo	$	393	99180.4
GF AT-16,1.2M,1M,80M,Mo	$	1394	351800.3
OFF AT-16,,1M,,Mo,Arcn.	$	948	239244.4
AT-16N,,1M,,Mo	$	518	130726.4
AT-16N,1.2M,1M,40M,Mo	$	1090	275080.6
terminale DTK			
DTK AT-16,1M,1.2,40M,Mo	A	10440	223063.6
DTK 386SX,1M,1.2Mf,noMO - 16Mhz,VGA,D-Top,noKey	A	9132	195116.6
- 20 Mhz,VGA,D-Top,noKey	A	10361	221375.7
DTK486-33M w 64K C.TOW - 4M,noW,FD,VGA,noMo,noKey	A	45077	963126.4
Calculatoare AT-386			
AT386,mainb,33,0M,nC	$	842	212493.4
386,12/25,1.2M,1M,40M,Mo	$	1526	385112.8
386,20/25,1.2M,2M,80M,Mo	$	2139	539814.1
CH.LT3600 16/20,40W,VLCD	$	3036	766187.7
LT.LA30,AT,1M,20W,MLCD	$	1960	494640.3
Monitoare			
12" mono FUJIT,VALTR	';U8.1		
14" mono SUPERTRON	$	110	27760.4
14" EGA SUPERTRON	$	407	102713.6
14" VGA (1024x768) SUPERT	$	414	104480.1
20" VGA monitor+cartela	A	86400	1846043.9
15" MAG allo A4	$	869	219307.4
Adaptoare grafice			
cartela mono	$	25	6309.2
cartela EGA	$	81	20441.8
VGA,8b.256k(800x600)	$	81	20441.8
VGA,16b,512k(1024x768)	$	138	34826.7
Subansamble			
FD 1.2Mb	$	81	20441.8
FD 1.44Mb	$	81	20441.8
FD contr. 2 driv. AT	$	25	6309.2
WD 1003 contr.	$	62	15646.8
controller SCSI	$	138	34826.7
Keyboard 101	$	46	11608.9

Denumire	(N)	Preţ (Valută)	Preţ (Lei)
GM6 mouse	$	21	5299.7
MICROSCAN	$	186	46940.4
COPROCESOR 287-8	$	168	42397.7
COPROCESOR 387-33	$	904	228140.2
MIDI tower+sursa	A	1747	37326.8
Discuri Winchester			
20 MB st225 (65ms)	$	233	58801.6
40 MB st251-1 (28ms)	$	331	83533.6
80 MB st4096 (28ms)	$	669	168833.9
1.2 GB CDC Winc.	$	5505	1389283.1
controller SCSI	$	138	34826.7
Memorii şi accesorii			
1 MB	$	66	16656.3
IC 41256-08	$	7	1766.6
Elemente de reţea			
ARCNET 8bit STAR-BUS	$	61	15394.4
Ethernet NE1000 8bit	$	138	34826.7
Etherlink 3COM 8bit	$	304	76719.7
Novell SFT 2.15	$	5654	1426885.8
Imprimante			
EPSON LX 800	D	235	35301.4
EPSON FX 1050	D	852	127986.4
EPSON LQ 2550	D	2760	414603.9
HP LaserJet III	$	2553	644294.2
HP LaserJet II/P	$	1604	404797.5
STAR NX1001	$	226	57035.1
Cartridge FX1050	$	5	1261.8
Surse neintreruptibile			
UPS 550 W	$	311	78486.3
APC UPS 1200 W	$	1228	309907.3
Altele			
Streamer Wangtech 60 MB IN	$	704	177666.7
Modem INT 1200/SM12H	$	94	23722.5
3.5" DS/DD floppy 3M	$	32	8075.8
3.5" DS/HD floppy 3M	$	40	10094.7
5.25" DS/DD floppy 3M	$	9	2271.3
5.25" DS/HD floppy 3M	$	16	4037.9
Leporello 240x1x12x1700	$	16	4037.9
Leporello 382x2x12x900	$	53	13375.5
Telefoane			
PHIL. D9026 (10 mem)	$	54	13640
CANON FAX 80	D	973	146162.9
SHARP 7850	$	2835	715461.9

Alternative pentru reţea

Multiuser-DOS produs de Digital Research este un sistem de operare multiuser, multitasking, compatibil DOS, creeat special pentru procesoarele Intel i386sx, i386 şi i486. Oferă soluţii multiuser ca într-o reţea şi suportă pînă la 64 de utilizatori. Conform datelor furnizate de Digital Reseach, fiecare utilizator poate executa maxim 8 aplicaţii distincte simultan la locul său de muncă. Prin capacitatea de "Dynamic Idle Detection" a Multiuser-Dos-ului, capacitatea de multitasking a procesorului este folosită mai bine. La consola sistem, Multiuser DOS lasă disponibili 576 Mbyte de memorie principală. Partiţiile de harddisk gestionate pot avea dimensiunea maximă de 2 Gbyte.

Informaţii: Digital Reseach, Englschalkinger Str. 14, 8000 München 81, tel. 089/9279940.

Compaq învingător la Las Vegas

Compaq Computer Corp. a obţinut în total 6 evidenţieri de la 3 reviste diferite; printre ele, titlul de "producătorul hard al anului" decernat de Systems Integration Magazine. Compaq a obţinut 4 locuri întîi la premiile acordate de această revistă.

La categoriile File Server şi sisteme multiuser, cîştigătorul a fost Compaq Systempro, iar la categoria portables/laptops cîştigător a fost Compaq LTE/286.

Editorii revistei PC Magazine au acordat sistemului Compaq Systempro locul întîi la categoria PC-uri, acesta devenind astfel cîştigătorul premiului "Technical Excellence" pentru 1990

În afară de aceasta, seria de produse Compaq Deskpro a fost aleasă de cititorii revistei "Reseller Management Magazine" ca "produsul care se vine cel mai bine" pe anul 1990.

Iată, din nou, cîteva preţuri orientative (aprilie 1991):

Sisteme	Preţ de stradă (no-name) (mărci)	Aparate de marcă, ca de ex.
AT		
80286 -CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy	1.299	Compaq 286 N1/SL cca 3.780
386SX		
80386SX-CPU, 1 Mbyte RAM, harddisk de 40 Mbyte, o unitate floppy, VGA-color	2.499	Compaq 386 N1/SL cca. 4.415
386/33		
80386-CPU, 2 Mbyte RAM, harddisk 150 Mbyte, o unitate floppy, VGA color	4.299	IBM-PS/2-80 A-31 cca. 13.350
486/25		
80486-CPU, 4 Mbyte RAM, harddisk de 320 Mbyte, o unitate floppy, VGA color	7.360	Compaq 486 cca 23.695

Conectare fără cablu

Un nou concept pentru transmisia de date între PC-uri precum şi pentru comanda unei imprimante fără cablu îl reprezintă "AC Datalink". Aparatele folosesc pentru transmisia de date reţeaua normală de alimentare. PC-uri şi echipamente periferice sînt legate prin intermediul prizelor. Comunicaţia poate să se desfăşoare independent de locul de amplasare la distanţe de pînă la 100 m. 7 PC-uri pot să-şi "împartă" o imprimantă - amplasamentul echipamentelor fiind indiferent. Viteza de transfer este de 840 bytes/secundă. Procedeul de verificare cu sumă de control de 22 biţi garantează o transmisie lipsită de erori.

Informaţii: Verran,Scheffelstr. 17, 3000 Hannover 1, tel. 0511/7083922

Advance Technology Award 1990

Pentru floppy-ul său de 3,5" cu o capacitate de 25 Mbyte, firma Insite a obţinut premiul "Corporate Advance Technology Award 1990" al clubului american al inventatorilor şi a fost primită în "Hall of Fame". Floptical-ul a fost declarat standard industrial pentru floppy-uri de capacitate mare încă la începutul anului trecut de către Micro Standards Committee.

Harddisk cu acceleraţie

"DC2030" se cheamă noul controller cache produs de Promise Technology Inc. pentru harddisk-uri realizate în tehnologie IDE (pentru bus de AT). Viteza de transfer se măreşte cu un factor între 3 şi 10, timpul de acces este de 0,5 ms (!). Controllerul poate controla de asemenea şi două unităţi floppy în formatele standard. În versiunea de bază, controller-ul dispune de 512 kbyte RAM; se poate face extensia on-board pînă la 4 Mbyte. O extensie opţională de memorie cache poate extinde capacitatea RAM-ului pînă la 512 Mbyte. Preţul controllerului în varianta de bază este de 980 mărci. Informaţii:
Atlantis Computer, Am Kirchplatz 7, 8133 Feldafing, tel. 08157/7051.

OS/2-ul a murit - trăiască OS/2!

La 28 ianuarie 1991, în pagina a treia, Wall Street Journal a anunţat că "in cadrul unei schimbări de curs mai largi, Microsoft va renunţa la OS/2".

Dacă ar exista un premiu de marketing în domeniul PC-urilor, acesta ar trebui acordat Microsoft-ului. Campania care ar trebui premiată s-a desfăşurat pe parcursul unui an întreg şi a ţinut în tensiune pe cei implicaţi cu 3 întrebări simple. În primăvara lui 1990, se punea întrebarea "OS/2 - urmaşul lui Windows?" În vară, întrebarea era: "Windows sau OS/2 ?" În toamnă, s-a început cu "OS/2-ul a murit ?" Această ultimă campanie este atît de "de efect" încît promite să ofere material pentru reviste pentru mai multe luni de zile.

La prima vedere se naşte impresia că Microsoft acţionează destul de debusolat. Întîi OS/2 era considerat urmaşul lui Windows, acum Windows se vinde ca urmaşul lui OS/2.

Microsoft ştia încă din 1984 că Unix-ul va fi o tentaţie pentru utilizatorii de DOS mai pretenţioşi. Încă de pe atunci, ţelul era :"nici un PC fără soft de la Microsoft". Deoarece 95% din PC-uri lucrau cu DOS, trebuia oferită o cale de dezvoltare care, faţă de "renaşterea" Unix-ului, să fie suficient de "tare" pentru a împiedica o migraţie masivă a utilizatorilor înspre Unix.

Dacă în 1987 nu s-ar fi lansat cu surle şi trîmbiţe OS/2-ul, Unix-ul ar avea o parte semnificativ mai mare din piaţă şi Microsoft-ul n-ar fi nici pe departe atît de mare. Deverul întreprinderii a fost, în 1987/1988, pe plan mondial, de 590,8 milioane dolari USA. La numai 2 ani după aceea, această cifră s-a dublat: 1,183 miliarde dolari.

Cînd a devenit evident că Apple nu prea poate face nimic serios pe cale judecătorească împotriva difuzării Windows, s-a lansat pe piaţă Windows 3.0. În primul trimestru al anului comercial 1990/1991, deverul realizat de Microsoft a crescut cu 77% - graţie lui Windows.

Cei care încă nu au înţeles niciodată OS/2 se vor bucura de presupusa veste despre dispariţia OS/2. Vor concluziona repede cît de multă dreptate au avut întotdeauna. Însă aceşti oameni încă nu au înţeles ce este un sistem de operare modern pentru PC. Vor fi în continuare de părere că Unix sau un Windows bazat pe DOS este "mai bun".

Îm mod cert însă, trei lucruri nu va face Microsoft-ul: să termine din proprie iniţiativă "vîlva" dimprejurul Windows-ului, să arunce codul pentru OS/2 sau să ofere Windows numai pe platforma DOS. Ce este mai aproape decît a vinde OS/2 în hainele Windows-ului? Ar trebui avut grijă doar ca programele existente sub OS/2 Presentation Manager să poată rula sub o versiune viitoare de Windows. Şi aceasta este ceea ce se va întîmpla.

)

i856 şi i846sx

Primele detalii despre proiectatul 80586 al Intel-ului circulă printre specialişti: 64 biţi intern şi extern, clock de 66 MHz, instrucţiuni multiple şi 386 compatibile, prelucrarea mai multor instrucţiuni în fiecare ciclu, cache de 16 kbyte, unitate de virgulă mobilă compatibilă 386 precum şi soft intern pentru diagnoză şi depanare. Compatibilitate soft pînă jos la 8088. Modul protejat este asemănător cu cel de la 386 şi 486. Performanţele unui 586 vor corespunde în mare cam de 4 ori celor ale unui 486 la 33 MHz.

Conform zvonurilor printre iniţiaţi, Intel pregăteşte o versiune redusă a lui 486, numită 486sx. Este o versiune de 486 fără coprocesor pe chip şi cu un preţ redus cu circa 200 $ faţă de versiunea "integrală". Se prevede un clock de 30 MHz. În noua versiune se vede şi o strategie de marketing faţă de strădaniile AMD în domeniul 386.

Orice AT 286 poate deveni un 386SX...

Intel oferă pe piaţă un aşanumit Snapln 386 Module. Modulul oferă tuturor PC-urilor bazate pe 80286 performanţele unui 80386SX (deci o dublare a performanţelor), prin simpla înlocuire a procesorului 80286 cu acest modul. Acelaşi efect se obţine la unele sisteme din seria PS/2. Prin aceasta, calculatoarele modificate pot prelucra şi soft de 32 biţi; mai ales aplicaţii de volum mai mare şi mai multe aplicaţii simultan. Şi Windows 3.0 poate fi rulat atunci în Enhanced Mode. Snapln 386 este prevăzut cu 16 kbyte memorie cache, drivere pentru DOS şi OS/2 şi cooperează bineînţeles cu 80287.

Computer Panoráma

Replică a revistei "Computer Persönlich" editată de Markt & Technik, revista maghiară "Computer Panoráma" oferă numeroase informaţii utile. Pentru cunoscătorii de limbă maghiară, redacţia revistei ne-a oferit posibilitatea de a vinde revista celor interesaţi.

Revista apare lunar, are circa 80 pagini, o ţinută grafică deosebită, numeroase "trucuri", listinguri - exemplu, informaţii la zi, reclame de specialitate , mică publicitate etc. Cu un preţ de 149 forinţi, revista poate fi abonată prin noi la un preţ (estimativ) de 250 lei.

Prietenul meu calculatorul

"Calculatoarele sînt proaste". Această sentinţă a grupului rock "Spliff" din Berlin are deja cîţiva ani vechime, dar este încă actuală.

Calculatoarele îi enervează pe utilizatori cu apeluri pedante de genul "Comandă sau nume fişier eronat" (Bad command or file name) - şi aceasta doar pentru simplul fapt că la comanda "Backup" s-a tastat un Y în loc de U.

Pentru a elimina această ignoranţă 60 de cercetători de la Societatea pentru Matematică şi Prelucrarea datelor - GMD ("Gesellschaft für Mathematik und Datenverarbeitung) din St. Augustin, lîngă Bonn, s-au pus pe treabă. Banalele greşeli de tastare sînt cele mai simple probleme cu care se confruntă. Principalul scop pe care şi l-au propus este realizarea "calculatorului asistent" care ar trebui să poată:

- corecta automat greşelile utilizatorului
- să se acomodeze obiceiurilor utilizatorului
- să ştie lucra şi singur fără să fie nevoie ca utilizatorul să intervină tot timpul.

Pe scurt: acest calculator ar trebui să fie ceva de genul unui asistent personal, care să "gîndească" singur, care să recunoască greşelile de formulare şi să le corecteze şi care să fie informat în permanenţă despre nevoile şefului său.

Ideea unui asistent electronic a provenit din supărările zilnice obişnuite în lucrul cu calculatorul. Oamenii de la GMD cunosc problema din proprie experienţă. Conducătorul Institutului de Informatică Aplicată al GMD, dr. Peter Hoschka spune: "de cîte ori nu ni se întîmplă să stăm în faţa monitorului şi să ne întrebăm, ce s-o mai fi întîmplat ?" Chiar şi utilizatorii rutinaţi îşi consumă încă prea mult timp pentru căutarea erorilor, pe care-ar putea să le elimine încă de la început cu asistentul electronic.

Pentru a dota calculatorul asistent cu aceste facilităţi fantastice, cercetătorii de la GMD fac apel la metodele inteligenţei artificiale (IA). Ei încearcă să facă calculatorul să execute lucruri pe care doar omul le execută pînă acum: acestea ar fi în primul rînd capacitatea de învăţare şi capacitatea de a recunoaşte greutăţile şi de a reacţiona pe măsură.

Aparatul de serie va trebui să înţeleagă corect chiar şi instrucţiuni neclare, vagi sau chiar presupuse. Încă un pas înainte, dacă ne gîndim că astăzi calculatoarele intră în grevă la o simplă greşeală obişnuită de tastare. Pentru a obţine acest lucru, calculatorul asistent are nevoie de un bagaj de cunoştinţe de bază; el trebuie să recunoască formularea imprecisă şi s-o corecteze amintindu-şi de o situaţie asemănătoare anterioară pe care s-o compare cu situaţia curentă.

Şi pentru cea de-a doua însuşire - facilitatea de a se acomoda obiceiurilor utilizatorului - sistemul viitorului are nevoie de o bază de cunoştiinţe pe care să şi-o poată extinde singur. El va observa utilizatorul ţinînd un jurnal al tuturor intrărilor, astfel încît - în funcţie de cît de mult s-a lucrat cu calculatorul - acesta să poată stabili un profil exact al utilizatorului. Vor fi stocate informaţii despre nevoile şi stilul de lucru al şefului. Pe baza acestor informaţii calculatorul se va putea acomoda nevoilor şefului său. Astfel calculatorul se va informa asupra a ceea ce utilizatorul ştie deja, va şti ce paşi nu sînt stăpîniţi încă şi va fi în stare să ajute.

Pentru că asistentul va trebui să rezolve singur multe sarcini, la dorinţă, va trebui să-şi poată face munca transparentă. În cazul ideal el va executa singur un întreg şir de paşi de lucru. În oriçe caz el va trebui să poată explica utilizatorului ceea ce face pentru ca acesta să nu aibă sentimentul că este subordonat calculatorului.

În sfîrşit cercetătorii aşteaptă de la intendentul mecanic, ca acesta să-şi cunoască cu exactitate competenţele. Dacă utilizatorul doreşte ceva ce depăşeşte limitele acestuia, el trebuie să fie în stare să explice unde îi este depăşită competenţa şi să ofere alternative.

În prezent GMD lucrează la trei realizări care ar trebui să demonstreze suportul inteligent oferit de asistentul electronic: designerul grafic, configuratorul de documente şi interpretorul statistic.

Un scop mai îndepărtat pentru designerul grafic ar fi, de ex., să se introducă în sistem doar o schiţă grosieră şi acesta să fie în stare să o transforme într-o imagine grafică estetică şi utilizabilă.

Interpretorul statistic trebuie să fie în stare să interpreteze singur materialul statistic livrat. Vor fi evaluate, de ex., încasările unui magazin sau rezultatele ultimelor alegeri. Pentru aceasta are nevoie, desigur, de cunoştinţele de bază necesare, trebuie să-i fie clar care declaraţii sînt interesante şi importante şi care interesează mai puţin. Pentru evaluarea rezultatelor alegerilor va trebui să cunoască în plus care sînt regulile după care trebuie evaluate mostrele ("Comparînd rezultatele alegerilor din '92 cu cele ale alegerilor din '90 se poate trage concluzia că alegătorii FSN au optat pentru PSM, sau doar că nu s-au prezentat la vot ?").

Dar acestea nu sînt nici pe de parte toate planurile celor de la GMD în ceea ce priveşte asistentul electronic: asistenţa mediilor (de informare), sistemele conversaţionale şi sistemele expert organizatorice sînt sisteme care ar putea fi de ajutor utilizatorului în comunicaţia şi cooperarea zilnică.

Asistenţa mediilor va fi în stare să ordoneze toate ştirile, indiferent că vin prin poştă, fax, sau alte medii electronice, să desluşească contextul şi să facă propuneri pentru o prelucrare ulterioară a lor.

O versiune preliminară a asistentului mediilor este deja testată practic în institut. Ea conţine o tratare clasică a adreselor, cu ajutorul căreia utilizatorul poate accesa diferite medii de comunicare.

La o simplă apăsare de tastă asistentul mediilor trimite o scrisoare la fax-ul conectat şi prin acesta la destinatarul dorit. Omul economiseşte astfel tot stres-ul cu formarea numărului, repetarea apelului sau distorsiunile de transmisie. Aceste munci ingrate le va prelua în viitor calculatorul, care va insista pînă cînd o sarcină va fi rezolvată în mod corespunzător. Acelaşi lucru este valabil şi pentru comunicările care se trimit prin reţea sau telefon.

În spatele complicatului denominativ "sistem expert organizatoric" se ascunde ceva simplu: calculatorul este îndopat cu toate informaţiile imaginabile despre organizarea unei manifestări sau excursii. El ştie unde poate rezerva camere la hotel, rezerva bilete la avion sau organiza săli pentru expuneri. Pregătirile unui congres devin astfel o joacă de copii.

Utilizatorul trebuie să-i spună doar ce grupă de persoane trebuie să participe (de ex. toţi proprietarii de gogoşerii din Piaţa Trandafirilor al căror nume începe cu o literă situată între A şi K) - tot restul rezolvă calculatorul. El ştie unde locuiesc respectivele persoane, unde le poate rezerva hotelul cel mai convenabil şi care sînt mijloacele de transport cu care pot ajunge în modul cel mai comod posibil la locul manifestării.

Fără să ceară nimic din partea utilizatorului, calculatorul trimite un fax la hotel pentru a vedea dacă pot fi puse la dispoziţie un număr suficient de săli pentru expuneri şi de camere la hotel, face rezervările la avion sau la tren şi trimite invitaţiile. Sfătuitorul electronic ştie chiar şi dacă în sălile de expunere există un diaproiector, epidiascop, copiator electrostatic, etc.

Întrebarea care-i frămîntă pe cei 60 de cercetători de la GMD este cum să ajungă informaţiile de bază în calculator. "Unul din locurile cele mai înguste de care ne lovim este achiziţia de cunoştinţe", mărturiseşte deschis dr. Hoschka. Deoarece calculatorul nu poate interpreta o situaţie în acelaşi mod în care o face omul. O condiţie trebuie formulată astfel încît ea să producă un rezultat fără echivoc ("dacă hotelul este ocupat, caută cel mai apropiat hotel şi reia procedura acolo").

Programatorii au deci nevoie de un expert pentru domeniul respectiv, care să fie în stare să formuleze toate regulile astfel încît calculatorul să se poată descurca cu ele. Chiar şi baza de date (în cazul hotelurilor toate adresele) trebuie organizată astfel încît calculatorul să poată găsi ceea ce îl interesează.

Cunoştinţele unui expert trebuie mai întîi valorificate - în jargonul IA "extrase". Această extracţie a cunoştinţelor se dovedeşte a fi în IA cel mai mare obstacol. Din acest motiv un întreg proiect se ocupă doar cu această întrebare. Pentru soluţionarea ei lucrează împreună un expert şi un "inginer de cunoştinţe". Expertul furnizează structura informaţiilor, iar cercetătorul IA introduce reprezentarea cunoştinţelor în calculator, fie sub forma unei baze de date, fie în altă formă.

Un alt proiect aflat în lucru la GMD este "asistentul activităţilor". El trebuie să coordoneze munca în colectiv a mai multor oameni, să împartă sarcinile, să le memoreze şi să le verifice şi să aibă grijă să se realizeze o cooperare fără incidente.

Toate facilităţile enumerate pînă acum sună foarte frumos, dar va exista în curînd un calculator cu o putere uriaşă de calcul care să gîndească alături de utilizator ? Va deveni străvechiul vis al cercetătorilor de a obţine inteligenţă sintetică realitate ?
Sau visele cercetătorilor nu sînt decît vorbe goale ?

Clar este că "aproape tot ceea ce s-a amintit pînă acum încă nu poate fi realizat azi. Altfel nici nu ar mai fi o preocupare a cercetătorilor" spune dr. Hoschka. Pentru a transpune ideile în realitate informatica trebuie să mai progreseze mult.

Cercetările în acest domeniu încă se află abia la început. Şi chiar la sfîrşitul celor 10 ani acordaţi proiectului pilot tot nu va exista încă un calculator asistent în producţie de serie. Scopul este în primul rînd "de a pune la punct bazele pentru a putea construi astfel de sisteme care să poată da asistenţă". Cine speră însă ca încă de mîine să-şi poată dota biroul cu un astfel de calculator acela se bucură înainte de vreme.

Cercetătorii sînt optimişti. Dar nu numai ei sînt cei care-şi pun speranţe în aceste proiecte ci şi cei care le finanţează - instituţii de stat (BMFT - Bundesministerium für Forschung und Technologie) şi particulare (Olivetti).

(R. M.)

IASTED
- organizaţie ştiinţifică internaţională accesibilă şi specialiştilor români

IASTED - The International Association of Science and Technology for Development - este o asociaţie internaţională înfiinţată în 1977, de organizaţii şi membrii individuali interesaţi de dezvoltarea economică cu ajutorul ştiinţei şi tehnologiei. Entitatea conducătoare este consiliul său executiv, sprijinit de cîteva comitete internaţionale formate din personalităţi ştiinţifice în domeniile de interes. Actualmente preşedintele IASTED-ului este profesorul M.H.Hamza din Canada.

Scopul principal al IASTED-ului este dezvoltarea economică viitoare cu ajutorul ştiinţei şi tehnologiei. O deosebită atenţie este dată zonelor mai puţin dezvoltate economic ale globului. Activităţile IASTED-ului includ schimbul şi circulaţia informaţiei în ştiinţă şi tehnologie, în cooperare cu organizaţiile naţionale şi private, organizarea de simpozioane, conferinţe, cursuri şi expoziţii internaţionale, publicarea de reviste, jurnale şi cărţi, şi consultaţii de specialitate. Zonele de interes în activităţile IASTED-ului sînt în general interdisciplinare.

Pînă acum, au fost organizate conferinţe pe teme diverse cum ar fi: simulare, modelare, energie, gestiune, birotica, control, electronica şi robotica. Conferinţele au avut loc în Zürich, Montreal, San Francisco, Atena, Paris, Cairo, Düsseldorf, Boston, Tunis. Innsbruck, Copenhaga, Brisbane şi Xian.

La ultimele conferinţe organizate în luna februarie în Austria a luat parte şi un grup de specialişti români de la Institutul de Cercetări în Informatică care au prezentat 2 lucrări: "Pachete software pentru controlul de calitate nedistructiv al produselor metalice unidimensionale" şi "ANTIS - un sistem integrat de testare şi sortare a ţevilor". Temele simpozioanelor au fost "Informatica Aplicată" şi "Modelare, control şi simulare". Următoarea conferinţă va avea loc între 10-13 Septembrie la Trani, Italia şi va avea ca subiect "Parallel Computing". În general taxa de participare la o conferinţă este între 400-500 $.

Periodicele editate de către IASTED sînt:

- -**Control and Computers**-care publică articole extrase din cărţi, notiţe de conferinţe şi anunţuri pentru manifestări ştiiţifice internaţionale, pentru subiecte legate de Teoria automatizărilor şi aplicaţii, Control şi Calculatoare.
- -**Sisteme Energetice** ce vizează toate aspectele producerii energiei din surse convenţionale sau neconvenţionale, dezvoltarea, proiectarea şi optimizarea reţelelor de transport şi distribuţie, valorificarea apelor geotermale, valorificarea energiei valurilor, etc.
- -**Jurnalul Internaţional de Modelare şi Simulare** - ce publică materiale despre limbaje, hardware, software, metodologii, identificare, metode numerice, metode grafice, VLSI, microcalculatoare în simulare, arhitecturi paralele, şi aplicaţii din toate domeniile.
- -**Robotics and Automation** - care este un adevărat jurnal internaţional de sisteme, control, simulare, inteligenţă artificială, senzori, controlere, procesarea imaginii, analiza economică, roboţi industriali, etc. O deosebită atenţie se acordă aplicaţiilor. Revistele şi jurnalele apar în general de trei sau patru ori pe an.

Opţiunea apartenenţei la această asociaţie este deschisă organizaţiilor şi persoanelor individuale interesate de ştiinţă şi tehnologie care au un minim de cerinţe satisfăcute. Taxa ce trebuie plătită în acest caz este de 60 $ sau 120 SFr pentru persoanele individuale şi respectiv 100 $ sau 200 SFr pentru instituţii. Ca urmare a plătirii acestei taxe se obţine şi un abonament pentru 1 an la una din publicaţiile IASTED. Membrii IASTED- ului beneficiază de asemenea de anunţuri privind toate reuniunile internaţionale organizate de IASTED şi de reduceri la taxa de participare, ei au satisfacţia de a fi membrii unei asociaţii ştiinţifice internaţionale la cel mai înalt nivel. Participînd la manifestările organizate de IASTED, se pot primi informaţii de prima mînă despre activitatea de cercetare curentă şi activităţile tehnice de aplicare a lor, de asemenea apare ocazia de a face cunoştinţă cu specialişti din alte ţări din domeniul propriu de activitate. Cei ce doresc să ia contact cu secretariatul IASTED se pot adresa la P.O.Box 25, Station G, Calgary, Alberta, Canada T31 2G1.

Avantajele apartenenţei la o societate sau asociaţie internaţională în legătură cu specificul activităţii proprii este evidentă oricărui specialist. A venit poate momentul ca şi specialiştii din România să le cunoască şi să poată participa la activităţile lor.

(ing. fiz. Luminiţa Todor)

Ce (şi de unde) putem cumpăra

Trei au fost întrebările cu care ne-am adresat interlocutorilor noştri:

1) Care este "capul de afiş", vedeta, cel mai ... dintre produsele pe care le oferiţi spre vînzare ("Rolls Royce-ul")?

2) Care ar fi produsul ofertat de d-voastră pe care l-aţi recomanda unui prieten bun cu recomandarea "ia-l repede, cît mai am, că e atît de bun la cît de ieftin este încît aşa ceva nu mai găseşti nicăieri"?

3) Care este produsul pe care l-aţi vîndut cel mai mult, şi, dacă nu este secret, cam în ce ordin de mărime?

De data aceasta ne-am prezentat ca fiind de la "if" şi am specificat şi faptul că răspunsurile urmează să le publicăm. Oricum, nu ne asumăm nici un fel de responsabilitate asupra datelor furnizate, ele fiind preluate pur şi simplu după cum ni s-a răspuns la telefon - uneori pe loc, uneori după un timp de gândire. Din nou, am baleiat firmele de a căror existenţă am aflat. Ne pare rău dacă am scăpat pe cineva - lista noastră este exhaustivă în ce priveşte firmele pe care le cunoaştem; ne pare sincer rău dacă am uitat pe cineva sau am nedreptăţit prin omisiune. Ne vom putea corecta desigur în alte viitoare anchete; ne exprimăm de pe acum gratitudinea tuturor celor care ne vor sprijini în a face cît mai cuprinzătoare şi mai reprezentative aceste anchete, pe care le dorim periodice şi cît mai interesante pentru cititorii noştri. Am făcut ancheta aceasta între 11 şi 16 aprilie 1991; mai lipsesc din ancheta noastră unele firme, pe care nu am reuşit să le contactăm în perioada susmenţionată.

(I.F.)

Şi iată acum răspunsurile (în ordinea alfabetică a firmelor):

AbMod s.n.c. **951/56.607 , 47386** **3400 Cluj, str. Minerilor nr. 42**

1) staţie grafică cu 386, harddisk de 80 Mbyte, VGA color	499.000 lei
+ coprocesor 387	100.000 lei
+ scanner Microtech MS-II	299.000 lei
+ placă captare video	200.000 lei
+ imprimantă Laser Star 82	399.000 lei
	total 1.497.000 lei
2) filtru monitor Polaroid CP - Universal II, (10" - 15")	15.000 lei
3)AT 286, harddisk de 40 Mbyte, VGA color	cca. 20/lună

ABTECH **90/10.42.73 , 10.09.65** **Bucureşti, sector 2, Calea Moşilor 201, bloc 9, sc. B, et. 7, apt. 53**

1) PC bazat pe i486, 16 Mbyte RAM, compatibil MS-DOS, OS/2, Novell, cîte un floppy de 3,5 şi respectiv 5,25", controller 2H2F (pote controla două harddisk-uri şi două unităţi de floppy), interleave 1:1, monitor Super VGA 1024 x 768, interfeţe: 2 x serial + 1 x paralel + 1 x game, clock/calendar battery backup, streamer 150 Mbyte, superquality enhanced 101 keyboard, harddisk de 760 Mbyte, modem intern la 2400 Baud	2.000.000 lei
2) AT-uri 286 cu monitor monocrom	470.000 lei
3) reţele (8+1, 16+1 sau 48+1); File server bazat pe 386, harddisk de 338 Mbyte, 4 Mbyte RAM, floppy 1 x 3,5" + 1 x 5,25", streamer de 60/120 Mbyte, monitor monocrom VGA; staţiile de lucru: 286 cu floppy 1 x 3,5" + 1 x 5,25", 1 Mbyte RAM, monitor Super VGA color; imprimante matriceale Star şi imprimante cu Laser	4 reţele

ADISAN **90/15.81.65, 14.54.55** **Bucureşti, str. arh. Ion Mincu 11**

1) AST 286, 12 MHz, 1 Mbyte RAM, harddisk 40 Mbyte, VGA color, interfeţe: 2 x serial + 1 x paralel, floppy: 1 x 5,25", sau respectiv HP Bell cu harddisk de 80 Mbyte şi în rest ca mai sus
2) Compaq 286/16 MHz, în rest ca la AST
3) Hyundai 286 8/12 MHz, în rest ca la AST

ADMEX **90/69.20.52**

1) AT 286, cu VGA color şi harddisk de 40 MHz	430.000 lei
2) - la înţelegere -	
3) calculatoare din tipul descris la pct.1	17 calculatoare

AGEXCOM International **90/65.40.03** **Bucureşti, C.P. 30-82**

1) Bondwell Laptop, AT 12MHz, 1 Mbyte RAM, monitor LCD, harddisk 20Mbyte, floppy: 1 x 3,5", acumulator, mouse, adaptor reţea şi 17 produse soft (printre care WS 5.5)	350.000
2) TCR Computer Division, (USA), 286/12MHz, harddisk 40 Mbyte, floppy 1 x 3,5" + 1 x 5,25", 1 Mbyte RAM, VGA color	300.000
3) calculatoarele de la pct. 2)	- e confidenţial -

ALLDROM S.R.L. **90/33.12.50** **Bucureşti, str. Ceaikovski 13, sc. A, apt. 1, sect. 2**

1)reţele locale la cheie;	
fileserver cu 386/33 MHz, harddisk de 1 Gbyte/15 ms, floppy 1 x 5,25" + 1 x 3,5", plăci Everex Ethernet Novell 386, Super VGA sau NEC 5D	2.800 - 5.000 $
workstation 286 minitower, harddisk de 40 Mbyte, floppy 1 x 5,25" + 1 x 3,5", VGA 800 x 600, 1 Mbyte RAM	1.360 $
Novell SFT 2.15 ptr. 100 utilizatori	2.400 $
2)imprimanta Epson EPL 7000, (laser), 6 pag./minut, 4 Mbyte RAM, > 30 fonturi	1.500 $
3) AT 286 (MBL, S.U.A.)	

		340.000 lei, cca 200 buc.
ARIS ELECTRONIC S.R.L.	**90/59.52.70, 59.38.94**	**Bucureşti, Calea Victoriei 135, etaj 7**
1) 386/33, Hitachi (Germania), 64 kbyte cache, 2 Mbyte RAM, floppy 1 x 5,25", harddisk Fujitsu 90 Mbyte/25ms, interfeţe: 2 serial, 1 parall, fără monitor		615.000 lei
2)286/25 MHz, harddisk 40 Mbyte, floppy, Super VGA 1024 x 768, recomandat ptr. DTP, cu monitor color GoldStar		425.000 lei
3)AT 286 în structură "standard", 12 MHz, harddisl 40 Mbyte, VGA mono		270.000 lei, pînă în 100 bucăţi
Attigram	**90/33.13.10**	**Bucureşti, str. Lisabona nr.7**
1) Amstrad PC-2386 - 4 Mbyte RAM, harddisk de 65 Mbyte, VGA color, 386/20 MHz, floppy 1 x 3,5" + 1 x 5,25" (HD), mouse incorporat		650.000 lei
2) XT (PC-1640) - 640 kbyte RAM, floppy 2 x 5,25" DD, EGA color, mouse incorporat, harddisk 30 Mbyte		250.000 lei
3) Amstrad PC-2286 - 1Mbyte RAM, harddisk 40 Mbyte, floppy 1 x 3,5" + 1 x 5,25", VGA color, mouse incorporat, 286/12 MHz		400.000 lei, cca. 150 bucăţi
Obs.: firma desface toate produsele Amstrad		
B.B. Computer	**966/73.412, 31.778**	**2900 Arad, str. Miron Costin sc.1, apt. 4**
1)AT 386 / 33 MHz, cu 80387, 4 Mbyte RAM, 64 kbyte cache, harddisk de 330 Mbyte/16 ms, floppy 1 x 5,25" (opţional + 1 x 3,5"), interfeţe: 2 x serial + 1 x par, Super VGA, opţional streamer, mouse GM6+, imprimantă HP Laserjet III		cca. 2.000.000 lei
2)AT 286 / 16 MHz, 1 Mbyte RAM, harddisk 89 Mbyte / 19 ms (ST 1102), floppy 1 x 5,25", interfeţe 2 x ser + 1 x par + 1 x game, Super VGA		<500.000 lei
3)AT 286 / 12MHz, 1 Mbyte RAM, harddisk de 40 Mbyte / 28 ms (ST 157A), floppy 1 x 5,25", Super VGA, imprimantă Epson FX 1050		550.000 lei
BroDeCo	**90/59.05.03**	**Bucureşti, Calea Dorobanţi 59, apt. 5**
1)AT 386/33 MHz, 4 Mbyte RAM, harddisk de 100 Mbyte, floppy 1 x 3,5" + 1 x 5,25", Super VGA color		1.430.000 lei
2)AT 386/25 MHz, 2 Mbyte RAM, harddisk de 80 Mbyte, floppy 1 x 3,5" + 1 x 5,25", VGA color		630.000 lei
3)AT 286/16 MHz, 1 Mbyte RAM, harddisk de 40 Mbyte, floppy 1 x 5,25", VGA mono, cu impr. Panasonic 80 col., 200 car./sec.		cca. 453.000 lei, cca. 60 calculatoare
CSI - Circle Systems International	**90/59.78.88 , 13.89.60, 13.83.81**	**71326 Bucureşti, str. Cîmpina 39,sect. 1**
1) (reţele Novell, toată gama, inclusiv aplicaţii;) Novell Netware 386 3.11, 100 utilizatori		7.500 $
2) compilatorul Turbo C++		70.200 lei
3) TCR 286 (S.U.A.), 12 MHz, floppy 1 x 5,25" + 1 x 3,5", 1 Mbyte RAM, harddisk 40 Mbyte, VGA color		430.000 lei, 1.000 - 1.500 bucăţi
DATACOM	**951/14.625**	**Cluj, str. Pietroasa nr. 11**
1) 386DX (20 MHz), harddisk 106 Mbyte, floppy 1 x 3,5" + 1 x 5,25", VGA color, monitor multisync color, cu coprocesor 387, 2 Mbyte RAM		800.000 lei
2) scanner de mînă Mars 800, cu softul aferent (Scankit)		100.000 lei
3) AT 286, harddisk 40/80 Mbyte, VGA color 256 kbyte, 1 floppy 5,25", 1 Mbyte RAM, 16 MHz		300.000 lei, între 50 şi 100 buc.
Electronic Design Consulting Group	**90/14.35.94, 21.26.44**	**76245 Bucureşt i 5, bd. Eroilor Sanitari nr. 27**
1) staţie grafică Tatung Mariner 4I, (SPARC), SUN compatibilă, poate lucra sub DOS sau Unix, 8 Mbyte RAM, extensibil pînă la 64, harddisk între 207 şi 654 Mbyte		5.000 - 10.000 $
2)Tatung TCS-7450, 80286/16 MHz, placă de bază cu controller de floppy şi harddisk, interfeţe: 2 seriale, 1 paralelă, 1 game, 1 Mbyte RAM, floppy 1 x 5,25"; separat placă VGA *800 x 600); inclusiv MS-DOS 4.01, fără harddisk şi monitor,		527 $
3)Tatung TCS-8916, 80386SX/16MHz, în rest ca la pct. 2		660 $
Elimpex	**90/50.57.77**	**Bucureşti, Piaţa Cosmonauţilor 7**
1)MBC - 19LXT, 386/33 MHz, 128 kbyte cache, 4 Mbyte RAM (max. 16), floppy 1 x 5,25", harddisk de 150 Mbyte / 18 ms, Multisync Super VGA color		15.000 DM
2) MBC 27 MT, 286/12 MHz, 1 Mbyte RAM, (max. 5), VGA color, harddisk 40 Mbyte, floppy 1 x 3,5"		4.000 DM
3)ca la pct. 2		
Euroshop S.R.L.	**90/15.77.63**	**Bucureşti, bd. Unirii nr.9, bl. 2A**
1) PC AT 486/25 MHz (Cytorn, Taiwan - dar prin Piaţa Comună), 2 Mbyte RAM, 64 kbyte cache, floppy 1 x 5,25", interfeţe 2 x ser + 1 x par, harddisk 660 Mbyte / 15 ms, SCSI		6.500 $
2) Cytorn 386 / 25 MHz, 2 Mbhyte RAM, floppy 1 x 5,25" + 1 x 3,5", harddisk 80 Mbyte, streamer 125 Mbyte, VGA color (512 kbyte), interfeţe 2 x ser + 1 x par, mouse, MS-DOS 4.01, baby case		850.000 lei
3) AT 286 / 12 MHz (Tokio Industrial), 1 Mbyte RAM, floppy 1 x 5,25" + 1 x 3,5", harddisk de 44 Mbyte / 18 ms, interfeţe 2 x ser + 1 x par, mouse, VGA color (512 kbyte), MS-DOS 4.01		400.000 lei 20 bucăţi
Feper	**90/87.67.14**	**Bucureşti, bd. D.Pompei nr.8, P.O.Box 3845**
Genesys Oradea	**991/36.235, 31516**	
1) AT PC 286, 1 Mbyte RAM, harddisk 40 Mbyte, floppy 1 x 5,25", VGA color, cu imprimantă Epson FX 1050, MS-DOS 4.01, două seturi de dischete 3M, softuri utilizator, aplicaţii		450.000 lei
2) - ca la pct. 1 -		
3) - ca la pct. 1 -		
Iiruc SA	**90/88.60.65, 88.20.70/173**	**72236 Bucureşti, bd. Dimitrie Pompei nr.6, sector 2**
Infocomp	**90/13.26.97**	**Bucureşti, bd. Mihail Kogălniceanu nr. 10, sector 5**
1) imprimante matriceale Mannesmann/Tally şi Epson, 136 col, 24 ace, resp. 80 col, 9 ace		110.000 resp. 50.000 lei
2) Databank Computer (Freemont, California) - AT 286, harddisk 40 Mbyte, floppy: 2 x 3,5", 1 Mbyte RAM, VGA color, baby case, Legal Bios (comp. Novell)		1120$
3) - ca la pct. 2 -		
INFOGO	**966/15.003**	**2900 Arad, str. Lucian Blaga 2-4**
1) -		

2) Mifix- Factura - Personal - Salarizare, set complet de programe aplicative de gestiune — 75.000 lei / apl.
3) cele de la 2) — de 5 ori în Arad

Info Team **90/14.93.98** **Bucureşti, str. Smirdan nr. 18, et. 1, apt. 15**
1) Wordstar (4.0) cu caractere româneşti, pentru orice imprimantă — 20.000 lei/instalare
2) -
3) -

KT - TECHNOLOGY ROMANIA Ltd. **90/88.33.40** **Bucureşti , str. Luterană 11**
1) Know-how-ul pe instalare reţele; servicii complete, aplicaţii la cheie
2) AT 286, harddisk 40 Mbyte, VGA — <280.000 lei
3) - ca la pct. 2 - — cca. 800 bucăţi

LASER COMPUTER ROMANIA S.A. **90/35.70.70** **Bucureşti, str. Olari nr.35, sect. 2**
1)486/25 MHz, 128 kbyte cache, interfeţe: 2 x serial, 1 x paralel, 6 Mbyte RAM, floppy 1 x 3,5" + 1 x 5,25" HD, harddisk de 613 Mbyte/16 ms, SCSI controller, Magic Multisync Super VGA (1024 x 768) — 14.476 $, a 185 lei/$
2) se recomandă orice configuraţie, asamblarea făcîndu-se în ţară după doleanţele clientului
3)LASER AT3-286/16 MHz, 1 Mbyte RAM, harddisk de 45 Mbyte/28 ms, floppy 1 x 3,5" + 1 x 5,25", VGA color 14" — 2.613 $, a 185 lei/$

Obs.: în preţ sînt incluse şi MS-DOS 4.01, PC Shell 6.0 şi Windows 3.0; garanţie 1 an, suport hard-soft, eventual şcolarizare în cadrul unui contract cadru, la centrul propriu de pregătire, contracte avantajoase de întreţinere postgaranţie

Lecom s.r.l. **90/50.64.20** **Bucureşti, Calea Victoriei 126**
LOGIC **924/46.652** **Sibiu, Calea Dumbrăvii 21**
1) AT 486/25 MHz, (ASI -S.U.A.), multisync VGA, harddisk 1 Gbyte, 16 Mbyte RAM, floppy 1 x 5,25" + 1 x 3,5", mouse, placă reţea Ethernet — 1.500.000 lei
2)AT 286, (ASI), 12 /16 MHz, VGA color 256 kbyte, harddisk 40 Mbyte, floppy 1 x 5,25" + 1 x 3,5", mouse , 1 Mbyte RAM — 380.000 lei
3) ca la pct. 2 — peste 100

Lorandra - Private Company **921/67.329** **2200 Braşov, P.O. Box 5-44**
1) un pachet soft pentru calculul angrenajelor, pe o idee nouă pe plan mondial (se va breveta) — cca. 30.000 lei
2) AT 286 /16 MHz, cu 80287, 1 Mbyte RAM, harddisk 40 Mbyte, floppy 1 x 5,25', VGA color — 350.000 lei
3) ca la pct. 2 — în jur de 25

Meda Electronics **994/33116, 36.705** **Baia Mare, str. Transilvaniei 2A/1**
Micro ATCI **954/17024, 20521** **4300 Tîrgu Mureş, O.P. 1, C.P. 64**
1) file server: 386/33 MHz, 8Mbyte RAM, harddisk 320 Mbyte/10,5 ms, floppy 1 x 3,5" + 1 x 5,25", VGA mono, streamer 80 Mbyte — parte în reţea
2) XT 640 kbyte RAM, floppy 2 x 5,25", harddisk 20 Mbyte, Hercules — 250.000
3) AT 286/12 MHz, harddisk 40 Mbyte, 1 Mbyte RAM, EGA color, imprimantă Epson FX 1000 — 510.000 lei, peste 100

Microcomputer Services Ltd. **941/17.414,33534, 33538** **1100 Craiova, hotel Minerva, str. M. Kogălniceanu nr. 1**
1)reţele de calculatoare (cu file-server AT 486, AT 386 şi workstation AT 286, Turbo XT) şi staţii grafice de proiectare (AT 286, imprimante Amstrad, plotter format A3 Sekonic SPL, scanner)
2)reţele de calculatoare cu serviciile noastre de instruire, documentaţii tipărite aferente produselor program)
3) calculatoare PC AT 286 (harddisk 40 Mbyte, 1 Mbyte RAM, VGA color) şi PC /turbo XT (harddisk 20 Mbyte, 640 kbyte RAM, Hercules, 12 MHz)

Mons Medius **994/17.414**
RBM **90/14.39.29** **Bucureşti, str. Bibescu Vodă 1,O.P. 53**
RCD **90/88.31.90** **Bucureşti 2, str. Fabrica de Glucoză 15**

Romtrust s.r.l. **90/10.40.05,17.08.40; Ghimbav 44** **Bucureşti 1, str. Caragea Vodă 3; Ghimbav, str. Nouă 182, jud. Braşov**
1) AT 486/25 MHz, (Express Hong Kong), 4 Mbyte RAM, floppy 1 x 3,5" + 1 x 5,25", harddisk 80 Mbyte, Super VGA color — 965.000 lei
2) Express Economy 286 Model 2, 1 Mbyte RAM, loppy 1 x 5,25", hardddisk 40 Mbyte, Super VGA color — 263.500 lei
3) - ca la pct. 2 - — peste 100

SCOP **90/11.74.21** **Bucureşti, str. Polonă 86**
1) IBM PS/2-95-AKD, 486/33 MHz, 8 Mbyte RAM (72 ns), floppy 1 x 3,5" (opţional o a doua unitate), harddisk de 320 Mbyte/ 12,5 ms, SCSI, tehnologie microchannel, standard grafic XGA, suportă DOS, OS/2, AIX (Unix pentru PS/2) — Încă nu are preţ stabilit; a fost lansat în ianuarie în S.U.A.
2) AT 286 Everex, 2 Mbyte RAM, floppy 1 x 3,5" + 1 x 5,25", harddisk 40 Mbyte, Super VGA color, MS-DOS 3.30, interfeţe 2 x serial + 1 x paralel — 470.000 lei
3) AT 286 Techteam (Taiwan), 1 Mbyte RAM, Super VGA color, floppy 1x 3,5" + 1 x 5,25", harddisk 40 Mbyte — 380.000 lei

Serco **90/19.24.01** **70.268 Bucureşti, str. Luca Stroici 34**
SIS **90/11.53.26** **70131 Bucureşti, str. Batiştei 11**
1) AST Premium (U.S.A.) 486/33 MHz, floppy 1 x 3,5" + 1 x 5,25", harddisk 640 Mbyte / 16 ms, VGA color, imprimantă matriceală 132 caractere — 21.000 $
2) Everex AT 286 (U.S.A.), 1 Mbyte RAM, harddisk de 40 Mbyte, un floppy, VGA color — 1.280$
3) Taiwan AT 286/12 MHz, 1 Myte RAM, 1 floppy 5,25", harddisk 80 Mbyte, Super VGA color, imprimantă A3 (?) — 1300 $, cîteva zeci

System Plus **90/79.13.91,10.83.42,46.66.74,77.47.63** **Bucureşti, str. Zăgazului 10, apt. 25, sect.1**
1)AT 486/25 MHz (Dell, U.S.A.), 4 Mbyte RAM, harddisk de 80 Mbyte, Super VGA color 1024 x 760 — 9.000 $
sau respectiv MacIntosh FX, 4Mbyte RAM, harddisk de 80 Mbyte, monitor color 13" — 14.000 $
2) AT 386/20 MHz (Iverson), 1 Mbyte RAM, harddisk 40 Mbyte, VGA color — 2.500 $
3) AT 386SX/16 MHz, în rest ca la pct. 2 — 2.200 $, cîteva zeci

T.D. Flex **961/35254**
(exclusiv dischete)

Sînt SX-urile mai bune decît AT-urile ?

De opt ani deja inima AT-urilor a constituit-o procesorul Intel 80286. Totuşi timpul nu-i mai este favorabil: procesorul Intel 80386SX, cu şase ani mai tînăr, încearcă, cu consecvenţă, să-şi înlocuiască înaintaşul.

Cumpărătorilor de calculatoare le este greu: preţurile PC-urilor dotate cu Intel 80286 şi respectiv Intel 80386SX diferă foarte puţin. Din acest motiv nu este uşor să alegi. Ambele tipuri prezintă atît avantaje cît şi dezavantaje.

Procesorul 80286 a fost prezentat în 1982 şi constituie baza AT-urilor. El dispune de o magistrală (bus) de date de 16 biţi şi de o magistrală de adresare de 24 de biţi. Procesorul poate lucra în două moduri de lucru distincte: Real Adress Mode (mod real) şi Protected Mode (mod protejat). În mod real poate adresa o memorie de lucru (RAM) de maxim 1 MByte. În mod protejat poate adresa însă pînă la 16 MBytes. Această capacitate rezultă din mărimea magistralei de adresare (2^{24}=16777200).

Procesorul 80386 este deja sensibil mai modern. El a apărut pe piaţă în 1985 şi fiind un procesor pe 32 de biţi dispune de o magistrală de date de două ori mai lată. În plus cu magistrala sa de adresare de 32 de biţi poate adresa direct o memorie RAM de pînă la 4 GBytes. O caracteristică a acestuia este modul de lucru virtual (Virtual Mode) în care poate emula, în memoria sa, mai multe procesoare 8086. În acest caz în memoria sa mai multe programe pot fi rulate simultan (multitasking). La aceaşi frecvenţă de tact, intern, un 386 este de patru ori mai rapid decît un 286.

Saltul de la 286 la 386 este enorm, însă nu numai în ceea ce priveşte puterea, ci şi în ceea ce priveşte preţul. Periferia procesorului trebuie să fie croită pe lăţimea magistralei de date. Un 386 devine mai scump deci nu numai datorită procesorului, ci şi datortită mediului care-l înconjoară, cum ar fi de exemplu placa grafică sau controllerul de harddisk.

Intel a găsit în 1988 o soluţie de compromis, pentru utilizatorii care doreau o putere de calcul mai mare dar nu-şi puteau permite preţul, oferind procesorul 80386SX. Acest procesor este un hermafrodit care se descurcă şi cu periferia AT-urilor. Intern el lucrează ca un "386 adevărat" pe 32 de biţi, spre exterior dispunînd însă doar de o magistrală de 16 biţi. În acest mod, el este tot atît de rapid intern ca un 386, dar în accesarea periferiei şi memoriei el trebuie să comute înapoi pe 16 biţi.

Datele de 32 de biţi sînt "înjumătăţite" astfel şi "împinse" una după alta pe magistrală ca printr-o pîlnie. Această operaţie consumă, aşa cum era şi de aşteptat, timp. Din această cauză, în practică, un SX este cu cca. 10% mai lent decît un 386, la aceaşi frecvenţă de tact. O altă limitare o constituie faptul că magistrala de adresare a unui 386SX este doar de 24 de biţi şi deci nu poate adresa direct decît 16 MBytes de memorie.

La operaţiile de calcul interne un 386SX este de 1.8 pînă la de 2.0 ori mai rapid decît un 286. În legătură cu accesul extern, la memoria de lucru, sau la memoriile de masă, avantajul vitezei sporite dispare. Dar chiar şi atunci, un 386 în combinaţie cu un coprocesor 80387SX este sensibil mai rapid decît un 286 în combinaţie cu un coprocesor 80287.

Pînă nu demult avantajele vitezei sporite, fără coprocesor, erau pur teoretice, deoarece 386SX exista numai la o frecvenţă de tact de 16 MHz, în timp ce erau produse procesoare 286 la o frecvenţă de tact de 20 MHz şi chiar de 24 MHz. În aceste cazuri 286 era mai rapid, un motiv suficient pentru cumpărători de a-l prefera lui 386SX.

După lansarea pe piaţă a versiunii 386SX cu 20 MHz a dispărut însă şi această reţinere. Tactul ridicat duce la o sporire a vitezei de lucru cu pînă la 25 de procente. Trebuie avută grijă însă ca tactul să fie utilizat în mod corespunzător, deoarece aproape toţi ofertanţii îşi fac reclamă afirmînd că accesul la memorie se face fără cicluri de aşteptare. Totuşi nu orice procesor funcţionează într-adevăr fără cicluri de aşteptare. Aceasta poate face ca fiecare astfel de "pas nefolosit" să ducă la o scădere a vitezei cu 7 pînă la 8 procente. În acest fel se poate risipi foarte uşor avantajul cîştigat iniţial.

Cel mai important avantaj al lui 386SX însă este altul. Între-timp pe piaţă a apărut software care utilizează facilităţile de multitasking ale lui 386 sau 386SX. Cel mai cunoscut dintre acestea este Windows 3.0 al lui Microsoft, care oferă un mod de lucru extins pentru 386. În legătură cu un 386, limita de 640 KBytes, impusă de sistemul de operare DOS, poate fi depăşită putînd fi accesată direct o memorie de pînă la 16 MBytes fără EMS.

La acest capitol procesorul 286 este pus la stîlpul de tortură, fără speranţa că se va putea salva. 386SX însă poate depăşi limita fără probleme, arhitectura sa pe 32 de biţi fiind echivalentă cu cea a unui procesor 386. Din punctul de vedere al software-ului un 386SX nu este decît un 386 mai lent.

Acest truc nu este de fapt chiar nou, cum ar putea părea la prima vedere, deoarece acelaşi lucru s-a întîmplat şi cu procesorul 8088 la începutul seriei PC-urilor. Cu toate că procesorul 8086, procesor "cinstit" pe 16 biţi, a fost conceput mai întîi, din motive de costuri, IBM s-a decis să lanseze pe piaţă mai întîi procesorul 8088 care intern avea o arhitectură pe 16 biţi, extern lucra însă numai cu o magistrală de date de 8 biţi.

Procesorul 386SX va depăşi şi va înlocui în scurt timp "bătrînul" 286. Avînd preţul doar cu puţin mai mare şi o putere sporită, este de înţeles că se va impune.

Ca o concurenţă suplimentară, Intel a conceput deja un urmaş pentru procesorul 386SX, este vorba despre procesorul 80386SL. Cu mai multe funcţii integrate, un spaţiu de adresare de 32 MBytes şi suplimentar EMS 4.0, magistrală AT şi controller cache, acesta are perspective de reuşită frumoase. Încă din start, produs pentru un tact de 20 MHz, el este mai rapid decît un 386SX la acelaşi tact. El oferă, deci, nu numai mai multe funcţiuni, ci şi mai multă putere de calcul. Din aceste motive nu ne va veni prea greu să ne luăm adio de la 286.

(R. M.)

Salvarea datelor pe bandă magnetică - cum lucrează un streamer

Bazele tehnice ale salvării datelor

Disc jochey, adio! De cînd chiar şi în domeniul PC-urilor apar volume de date din ce în ce mai mari, unităţile de bandă magnetică Streamer s-au impus ca soluţii mai comode şi mai sigure pentru stocare (back-up) decît soluţiile cu dischete.

Mult timp salvarea datelor, de pe PC-uri, a fost de importanţă secundară. De obicei era de ajuns o soluţie soft, prin care programe back-up rezolvau salvarea/restaurarea datelor pe dischete, într-un mod mai mult sau mai puţin confortabil. Cît timp volumul datelor nu depăşea 20 MB era o soluţie bună, datorită costurilor mici. Astăzi însă la o capacitate uzuală a discului hard de 40 MB un back-up de pe dischete durează cel puţin un sfert de oră. Şi discurile hard ating capacităţi tot mai mari.

Acelaşi lucru este valabil şi în domeniul calculatoarelor mari. Cerinţele de capacitate de stocare a sistemelor de salvare/restaurare rezultă direct din dezvoltarea capacităţii discurilor. Dacă primele discuri Winchester aveau o capacitate de 70 MB la un diametru de 14 inchi, azi mărimea standard este de peste 1 GB la un diametru de 8 inchi.

Asemenea ordine de mărime, valabile în trecut doar la sistemele mari, au devenit curente, în termen de cîţiva ani, şi în domeniul calculatoarelor personale. Progresul legării în reţele a calculatoarelor duce la cerinţe şi mai mari de capacitate (salvarea de pe Server şi de pe discurile locale)

Dezvoltarea capacităţii de stocare pe benzi trebuie să poată

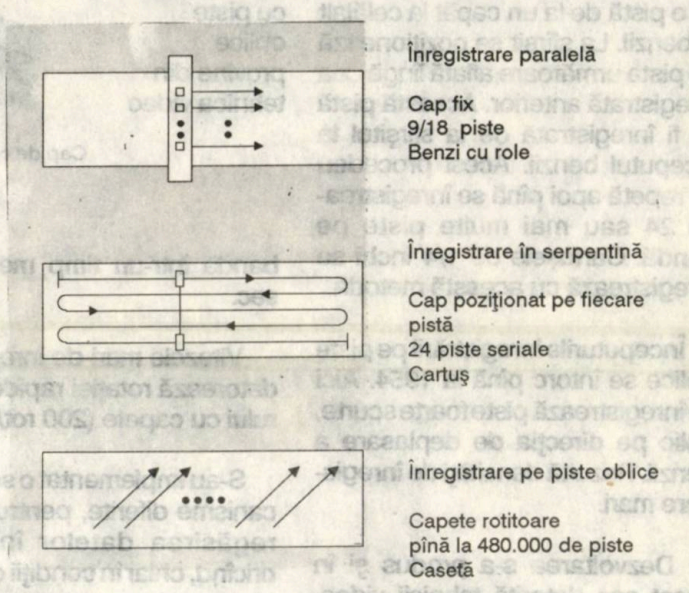


Fig. 1. Cele mai importante tehnici de înregistrare

răspunde capacităţilor de disc crescînde. Datorită tendinţei de minimalizare însă s-au micşorat şi dimensiunile mecanice ale mediilor de stocare.

Prima apărută din punct de vedere istoric, însă şi azi răspîndită, este unitatea de bandă cu role, pentru bandă de lăţime 1/2 inchi. Ea se constitue într-un standard de facto. Avînd diametrul maxim de rolă, banda pe suport subţire şi densitatea de înregistrare maximă, capacitatea benzii este totuşi limitată la o capacitate în jur de 200 MB.

Desprinderea de la principiul rolelor s-a realizat prin răspîndirea cartuşelor cu bandă de 1/2 inchi. Capacitatea acestora se află, în funcţie de numărul pistelor înregistrate, între 200 şi aproximativ 500 MB. Azi sînt foarte răspîndite cartuşele cu bandă de 1/4 inchi cu o capacitate de pînă la 525 MB.

Banda de 8 mm, folosită din 1985 la înmagazinarea datelor îşi are originea în tehnica video. Cu o capacitate de stocare de 2,2 GByte corespunde mai bine cerinţelor de astăzi de capacitate, decît rolele sau cartuşul.

Firma Gigatape, în 1987, folosind caseta DAT ca mediu de stocare, a ales pentru prima dată o tehnică de înregistrare în totalitate digitală. Caseta de mărimea unei cartele de cecuri şi ieftină este deja larg răspîndită în domeniul tehnicii audio (Digital Audio Tape). Gigatape a realizat cu ea un salt la o capacitate de 1 GByte; între timp au fost realizate creşteri considerabile ale capacităţii, aşa cum se va arăta în continuare, această tendinţă fiind inclusă şi în planurile de viitor.

Ştiinţa pistelor

Pe lîngă utilizarea de materiale diferite pentru benzi s-au dezvoltat şi au fost transpuse în practică şi tehnici de înregistrare diferite (Fig. 1).

Scrierea pe piste paralele este utilizată la benzile cu role de bandă de 1/2 inchi. În acest caz un cap fix înregistrează 9 sau 18 piste paralel şi în acelaşi timp în direcţie orizon-

tală pe bandă (paralelă cu muchiile).

La srierea în "serpentină" dezvoltată în 1972, capul scrie orizontal o pistă de la un capăt la celălalt al benzii. La sfîrşit se poziţionează pe pista următoare aflată lîngă cea înregistrată anterior. Această pistă va fi înregistrată de la sfîrşitul la începutul benzii. Acest procedeu se repetă apoi pînă se înregistrează 24 sau mai multe piste pe bandă. Cartuşele de 1/4 inchi se înregistrează cu această metodă.

Începuturile înregistrării pe piste oblice se întorc pînă la 1954. Aici se înregistrează piste foarte scurte, oblic pe direcţia de deplasare a benzii. Rezultă densităţi de înregistrare mari.

Dezvoltarea s-a produs şi în acest caz datorită tehnicii video. Sistemele VHS, Betamax şi Video-2000 lucrau deja cu înregistrare pe piste oblice (Generaţia 1), ca şi sistemele video mai noi pe 8 mm (Generaţia 2). Cu toate acestea prelucrarea complet digitală a semnalului s-a realizat prima dată la tehnologia DAT (a 3-a Generaţie a tehnicii de înregistrare pe piste oblice).

Frumos, oblic şi rapid

Tehnologia R-DST utilizează avantajele unităţilor de bandă DAT, deoarece le implementează împreună cu o electronică specială dezvoltată pentru scrierea datelor (Rotating bread Digital Storage Tape).

Banda va fi deplasată într-un unghi corespunzător (6 grade şi 23') cu o viteză absolută mai mică de-a lungul tamburului cu capete (Fig. 2). Banda înfăşoară tamburul cu capete doar 90 de grade (la sistemele de 8 mm 221 de grade) Prin aceasta pe de o parte banda va fi supusă unor solicitări mecanice mai mici; pe de altă parte este posibilă o viteză de căutare de 200 de ori mai mare ca la citire/scriere. Se poate accesa orice dată pe

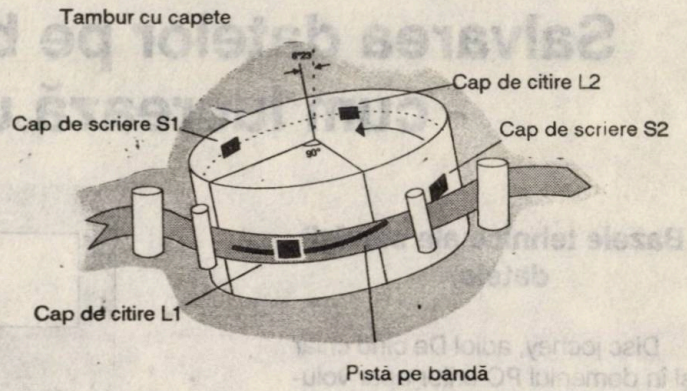


Fig. 2. Înregistrarea cu piste oblice provine din tehnica video

bandă într-un timp mediu de 20 sec.

Vitezele mari de înregistrare se datorează rotaţiei rapide a tamburului cu capete (200 rot/min).

S-au implementat o serie de mecanisme diferite, pentru a asigura regăsirea datelor înregistrate oricînd, chiar în condiţii defavorabile.

În timpul formatării benzii se recunosc porţiunile defecte de bandă, sînt marcate ca atare şi devin nedisponibile pentru utilizare în continuare. Pentru aceasta întreaga bandă este înregistrată cu o configuraţie specială de date care vor fi citite la sfîrşit, la o a doua trecere.

Acelaşi lucru este posibil prin Read After Write (RAW), adică citire după scriere fără o formatare care necesită timp. Condiţia pentru aceasta este un tambur cu cîte 2 capete de citire şi scriere (Fig. 2). Capul de citire L1 ce urmează capului de scriere S1 (după 270 de grade) citeşte configuraţia de date scrisă anterior. Dacă există diferenţe, atunci şi aici porţiunea de bandă defectă se poate exclude.

Siguranţa necesită spaţiu.

Metoda folosită de Gigatape de Forward Enor Conection (FEC) se remarcă prin faptul că datele nu mai sînt sortate şi corectate la nivel de biţi ci de octeţi. Pentru aceasta sînt generate şi înregistrate printr-o procedură specială (Reed - Solomon) pînă la trei coduri de corecţie. Datele şi octeţii de corecţie nu se succed pe bandă, nemijlocit. Datorită acestui interleaving (decalaj) se pot sări şi porţiuni mai mari de bandă defectă.

Atîta siguranţă îşi are preţul lui: datele de recunoaştere a erorilor şi codurile de corecţie necesită circa 30 % din capacitatea benzii. Dacă după această metodă se detectează o eroare prin RAW, atunci blocul de date respectiv poate fi scris, dacă este cazul de mai multe ori în

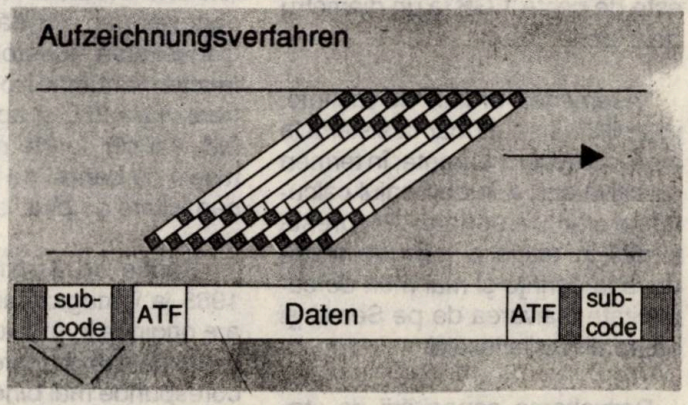


Fig. 3. Format de înregistrare pe piste oblice. Informaţiile ATF şi datele de corectură limitează domeniul de date.

alte zone de pe bandă. Doar cînd după citiri repetate rezultă că blocul a fost scris repetat greşit se va întrerupe procedura şi se va semnala eroarea. Deplasările de piste datorate mecanismului de antrenare sînt recunoscute electronic şi eliminate. ATF (Automatic Track Finding - regăsirea automată a pistei) ceea ce face de prisos reglajele mecanice.

Scriere cu format

Organizarea datelor ce urmează a fi scrise pe bandă are loc conform unor formate determinate. Fig. 3 permite o reprezentare a formatului de înregistrare pe piste oblice. În cadrul unei piste sînt disponibile pentru înregistrarea datelor 128 blocuri, fiecare de cîte 32 octeţi. În plus sînt înmagazinate în această zonă şi octeţi de detectare şi corecţie de erori. Informaţiile ATF (5 blocuri), zona Subcod (8 blocuri de date pentru organizarea benzii) ca şi semnalele de sincronizare (zona haşurată) se regăsesc atît la începutul cît şi la sfîrşitul unei piste.

Formatele care acum se află doar în curs de standardizare sînt Data/DAT şi DDS (Digital Data Storage). Acestea sînt prin definiţie incompatibile.

Pentru a permite utilizatorului citirea casetelor cu formate diferite cu aceaşi unitate, firma Gigatape a dezvoltat Multi-DAT. La aceasta formatul dorit se încarcă de pe o casetă format. Casetele de date scrise în formatul corespunzător pot fi astfel citite şi scrise în continuare. O schimbare de format este oricînd posibilă, sistemul este deschis pentru formate ulterioare.

Capacitate mai mare prin comprimare

Salvarea datelor fără intervenţia utilizatorului este posibilă doar atunci cînd capacitatea mediului pe care se face salvarea este mai mare decît volumul total de date de salvat.

Pentru cazurile cînd această premiză nu poate fi îndeplinită sînt furnizate aşa numitele Stacker (stivuitoare). În acest caz este vorba despre subsisteme care pot administra autonom un anumit număr de casete (de ex. 10). Un Stacker înlocuieşte o casetă plină cu una goală în mod automat. Scopul producătorilor de sisteme de stocare a datelor este totuşi urmărirea folosirii tuturor posibilităţilor de creştere a capacităţi casetelor.

O posibilitate de a realiza acest lucru este oferită de comprimarea datelor. Datele ce vin de la calculator sînt, înainte de a fi scrise pe bandă, comprimate după algoritmul Lempel Ziv. Prin acesta sînt înlocuite configuraţiile de caractere repetate prin combinaţii mai scurte de caractere (Ex: "GGGGGGGG" devine "8G"). Factorul de comprimare este totdeauna funcţie de natura datelor de comprimat (Fig. 4). Datele cu un număr mare de eşantioane de caractere redundante (de ex: datele de imagine) se pot comprima mai mult (factor 4,24) decît cele care conţin şiruri de caractere redundante mai puţine (date binare, factor 2,70).

Dacă presupunem că toate tipurile de date apar în practică la fel de des rezultă un factor de comprimare mediu de 3,26. Corespunzător cercetărilor proprii, Gigatape furnizează un factor de comprimare mediu de 2,5 (factor ponderat cu frecvenţa apariţiei). Factorul conţine şi o anumită rezervă.

Prin aceasta capacitatea unei casete DAT se ridică de la 1 GByte nominal la 2,5 GByte real. Prin aceasta creşte şi viteza de citire/scriere de la 150 KByte/s fizic la 300 KByte/s logic.

La sistemele Turbo DAT oferite de Gigatape comprimarea este implementată hard, adică nu sînt necesare operaţii utilizator în plus. Banda văzută logic a devenit, pur şi simplu, mai lungă. Cipuri speciale de comprimare oferă în plus faţă de cîştigul obişnuit în viteză o soluţie hard faţă de una soft.

O posibilitate la îndemînă pentru creşterea capacităţii este creşterea lungimii benzii. Dacă pînă acum se ofertau casete DAT cu lungime maximă a bezii de 60 m, acum, la începutul lui 1991, au apărut pe piaţă benzi de 90 m. Gigatape testează acum felul în care rezistă benzile cu suport mai subţire la solicitările mecanice într-o unitate DAT. Dezvoltările viitoare au ca scop o creştere în continuare a lungimii benzii ca şi o micşorare a lăţimii pistelor. Se pot prevede în scurt timp capacităţi de 10 GByte pe casete DAT. Tehnologia R-DST se arată de asemenea la nivelul viitoarelor dezvoltări.

(A.D.)

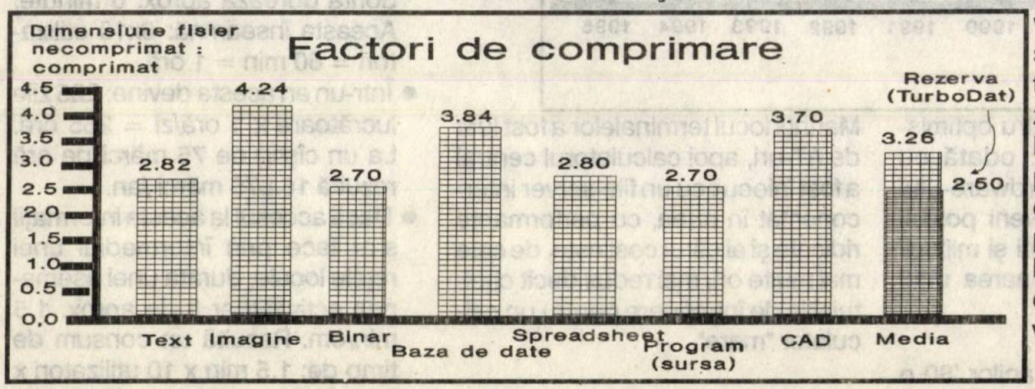
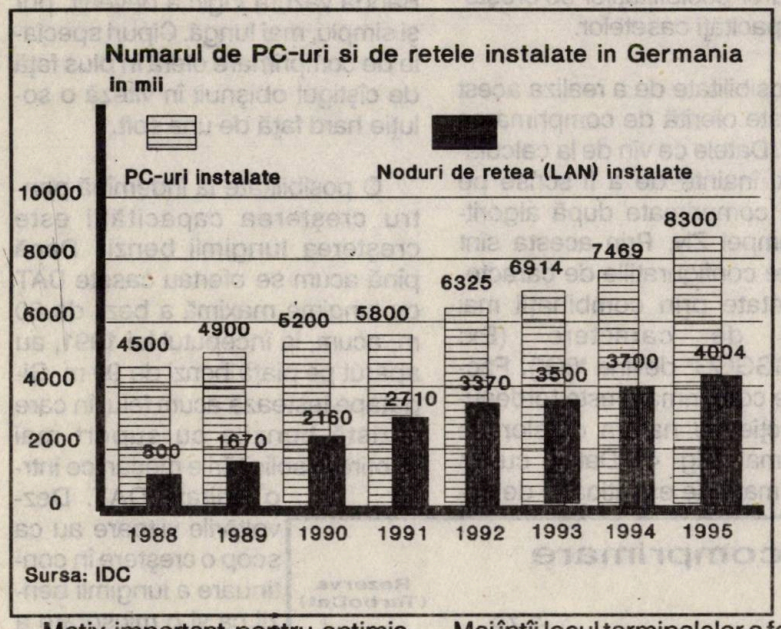


Fig. 4. Factorul de comprimare depinde de tipul datelor

Reţele de PC-uri la preţuri avantajoase

Din prognoza unui institut de specialitate - IDC din Germania, pentru testarea pieţii de calculatoare personale rezultă că în 2-3 ani tot al doilea PC va fi conectat într-o reţea locală (LAN). Analiştii pieţii de la un alt institut - Frost&Sullivan - apreciază că cifra de afaceri cu reţele de PC-uri va fi în anul 1994, în Germania, de o jumătate de miliard de mărci.

Pînă în anul 1995 vor exista pe teritoriul Germaniei aprox. 4 milioane de reţele - aproape tot atîtea cîte PC-uri şi AT-uri erau în folosinţă in anul 1987. Deci, după părerea institutului sus-amintit, acolo unde acum 4 ani exista un calculator personal izolat, în anul 1995 va fi instalată o reţea.



Pînă în 1995 tot al 2-lea PC va fi conectat într-o reţea

Motiv important pentru optimismul analiştilor pieţii: odată cu scăderea preţului hardware-ului calculatoarelor, va deveni posibil ca şi întreprinderile mici şi mijlocii să-şi permită achiziţionarea unei reţele.

Pînă pe la mijlocul anilor '80 o întreprindere care dorea să realizeze un acces comun la aceiaşi bază de date, de la mai multe posturi de lucru, trebuia să-şi cumpere un calculator central scump şi, în plus pentru fiecare post de lucru, cîte un terminal. Aceasta cerea un efort financiar de cîteva sute de mii de mărci. Pentru majoritatea firmelor mici această soluţie era de neabordat.

Între timp s-a ajuns ca şi un AT rapid să fie oferit la un preţ la care, în anii respectivi, se putea cumpăra doar un simplu terminal, compus din monitor şi tastatură.

Drept urmare firmele mari au început de cîţiva ani să-şi înlocuiască progresiv aplicaţiile pe calculatoare "mari" cu reţele locale, la fel de eficiente şi de performante. Mai întîi locul terminalelor a fost luat de AT-uri, apoi calculatorul central a fost înlocuit cu un file-server interconectat în reţea, cu performanţe ridicate şi al cărui cost este, de cele mai multe ori, mai redus decît cheltuielile de întreţinere pentru un calculator "mare".

Firmele mici şi mijlocii, au pornit pe drumul către reţele în alt mod: Mai întîi şi-au cumpărat PC-uri singulare, pe care şi-au relizat primele aplicaţii necesare. Într-o etapă următoare aceste PC-uri au fost interconectate în reţele locale, mai mari sau mai mici, în funcţie de necesităţi şi de posibilităţile financiare. Avantajele datorate flexibilităţii configurării acestor reţele, utilizării în comun a tuturor resurselor de calcul disponibile şi perspectivelor de dezvoltare au compensat din plin investiţia făcută.

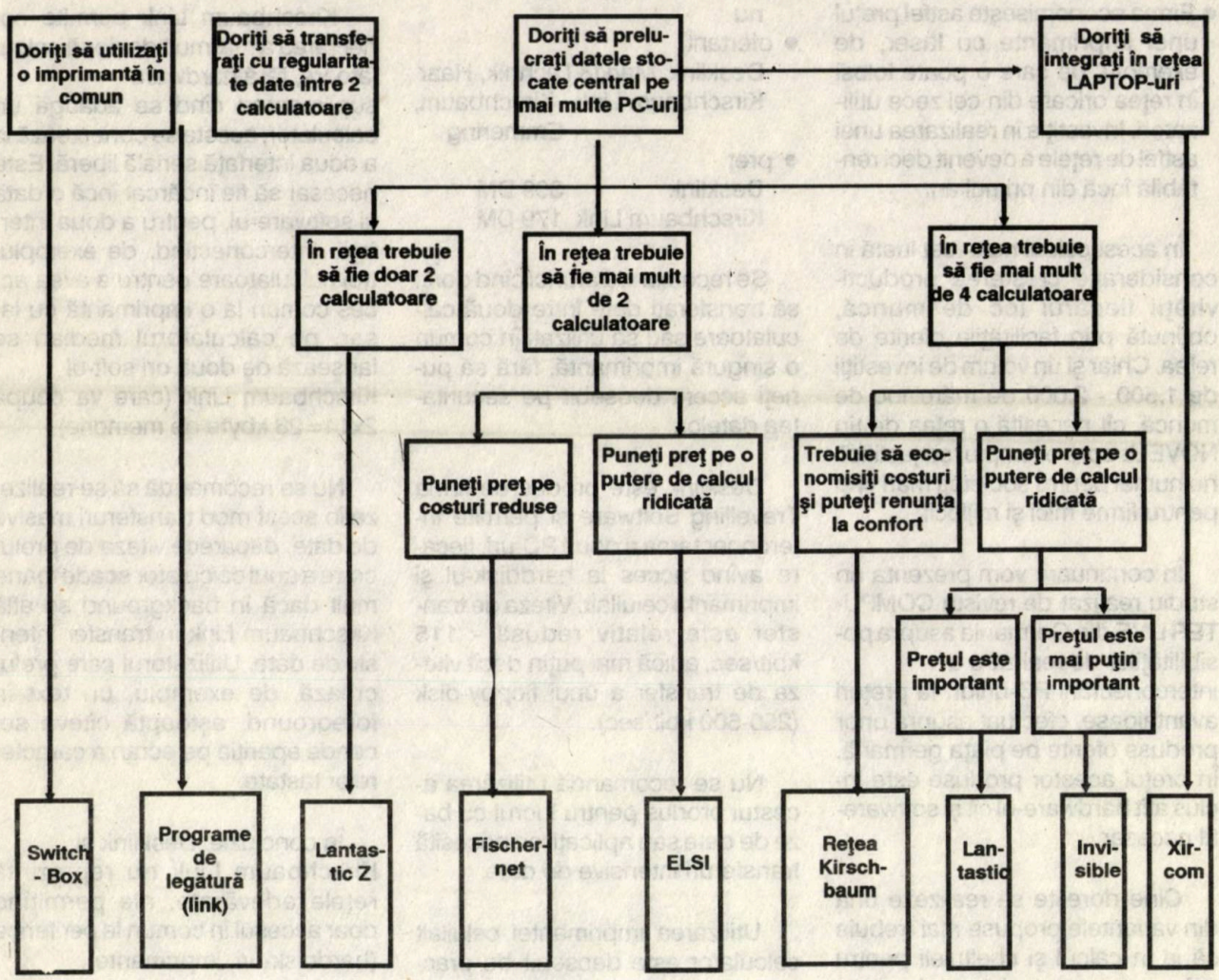
Să ilustrăm printr-un exemplu justificarea unei investiţii într-o reţea, chiar la o firmă mică:

- Pentru a realiza o reţea cu cele zece PC-uri existente la o societate de asigurări, sau la o agenţie de DTP, sînt necesare aprox. 10.000 - 15.000 mărci.

Să calculăm ce avantaje se obţin numai din timpul de lucru economisit:

- Presupunem că fiecare din cei zece angajaţi care utilizează un fişier de clienţi au nevoie numai o dată pe zi de o adresă de client din baza de date centrală. Fişierul de clienţi se găseşte pe PC-ul de la secretariat. Drumul pînă la secretariat şi aşteptarea pînă cînd secretara găseşte adresa dorită durează aprox. 6 minute. Aceasta înseamnă: 6x10 utilizatori = 60 min = 1 oră.
- Într-un an acesta devine: 225 zile lucrătoare x 1 oră/zi = 255 ore. La un cîştig de 75 mărci pe oră rezultă 16.875 mărci /an.
- Dacă accesul la aceste informaţii s-ar face prin intermediul unei reţele locale, durata unei asemenea activităţi ar fi de aprox. 1,5 min/om. Rezultă un consum de timp de: 1,5 min x 10 utilizatori x 225 zile = 3.375 min= 56,25 ore. Ceea ce înseamnă în bani:

Aşa alegeţi în mod corect:



1 Program de legătură (link)

Confortabil şi ieftin: Desk-Link şi Kirschbaum-Link - ideale pentru pretenţii simple

2 Lantastic Z

O reţea "curată", optimală pentru două PC-uri, fără hardware auxiliar.

3 Reţea Kirschbaum

Cea mai rentabilă intrare în lumea reţelelor - nu foarte confortabilă dar serioasă.

4 Fischernet

Un Starterkit pentru patru locuri de muncă; este excelent pentru şcolarizare cu grupe mici de cursanţi.

5 Lantastic

Multă reţea pentru bani puţini, o reţea cu confort ridicat şi o siguranţă ridicată a datelor, şi cu numeroase servicii auxiliare.

6 Invisible Ethernet

Foarte puternică şi "potrivită" pe caracteristicile sistem speciale ale staţiilor PC.

7 ELS I

Versiunea ieftină a gigantului LAN-urilor: Novell - poarta spre "lumea bună" a reţelelor.

8 Pocket - Adapter

Excelent chibzuit de producătorul profesional de reţele - Xircom din SUA, este folosit pentru cuplarea unui LAPTOP în reţea, nu necesită nici o mufă în PC.

56,25 x 75 = 4.218,75 mărci/an, adică o economie de 12.600 mărci anual.

- Firma economiseşte astfel preţul unei imprimante cu laser, de exemplu, pe care o poate folosi în reţea oricare din cei zece utilizatori. Investiţia în realizarea unei astfel de reţele a devenit deci rentabilă încă din primul an.

În acest calcul nu a fost luată în consideraţie creşterea productivităţii fiecărui loc de muncă, obţinută prin facilităţile oferite de reţea. Chiar şi un volum de investiţii de 1.500 - 2.000 de mărci/loc de muncă, cît necesită o reţea de tip NOVELL spre exemplu, se justifică nu numai pentru societăţi mari ci şi pentru firme mici şi mijlocii.

În continuare vom prezenta un studiu realizat de revista COMPUTER LIVE din Germania asupra posibilităţilor de realizare a interconectării PC-urilor, la preţuri avantajoase, efectuat asupra unor produse oferite pe piaţa germană. În preţul acestor produse este inclus atît hardware-ul cît şi software-ul necesar

Cine doreşte să realizeze una din variantele propuse mai trebuie să ia în calcul şi cheltuieli pentru asigurarea unui sistem de siguranţă a funcţionării reţelei (sursă neîntreruptibilă pentru căderi de tensiune, streamer pentru salvarea datelor importante, etc.).

1. Desklink - Kirschbaum Link

Date tehnice:

- posturi de lucru
  Desklink 2
  Kirschbaum Link oricîte
- utilizare server ca post de lucru
  fiecare PC este simultan server şi post de lucru
- necesar memorie pe server
  Desklink 30 - 70 kbyte
  Kirschbaum Link 14 kbyte pentru fiecare interfaţă
- hardware necesar:
  PC/XT/AT/386 cu MS-DOS
  interfeţe seriale
- software de reţea admis:
  nu
- ofertant
  Desklink Markt&Technik, Haar
  Kirschbaum Link - Kirschbaum, Emmering
- preţ
  Desklink 399 DM
  Kirschbaum Link 179 DM

Se recomandă atunci cînd doriţi să transferaţi date între două calculatoare sau să utilizaţi în comun o singură imprimantă, fără să puneţi accent deosebit pe securitatea datelor.

Desklink este produs de firma Travelling Software şi permite interconectarea a două PC-uri, fiecare avînd acces la harddisk-ul şi imprimanta celuilalt. Viteza de transfer este relativ redusă - 115 kbit/sec, adică mai puţin decît viteza de transfer a unui floppy-disk (250-500 kbit/sec).

Nu se recomandă utilizarea acestui produs pentru lucrul cu baze de date sau aplicaţii ce necesită transferuri intensive de date.

Utilizarea imprimantei celuilalt calculator este deosebit de practică, deoarece aveţi acces la ea direct din programul de aplicaţie. Dacă unul din calculatoare este conectat într-o reţea locală, atunci aveţi acces şi la imprimantele reţelei.

Desklink necesită aprox. 70 kbyte de memorie, iar în cazul în care dispuneţi şi de memorie EMS, spaţiul ocupat în cadrul celor 640 kbyte MS-DOS se reduce la 30 kbyte.

Kirschbaum Link oferă, ca şi Desklink, posibilitatea accesului la periferice celuilalt calculator, iar versiunea Plus conţine şi o suprafaţă utilizator cu un administrator de fişiere confortabil. Acesta permite vizualizarea, redenumirea, schimbarea atributelor, copierea fişierelor între cele două calculatoare.

Kirschbaum Link permite conectarea a mai mult de două calculatoare, fără hardware suplimentar: cînd se adaugă un calculator, acesta se conectează la a doua interfaţă serială liberă. Este necesar să fie încărcat încă o dată şi software-ul, pentru a doua interfaţă. Interconectînd, de exemplu, trei calculatoare pentru a avea acces comun la o imprimantă cu laser, pe calculatorul median se lansează de două ori soft-ul Kirschbaum Link (care va ocupa 2x14=28 kbyte de memorie).

Nu se recomandă să se realizeze în acest mod transferuri masive de date, deoarece viteza de prelucrare a unui calculator scade foarte mult dacă în background se află Kirschbaum Link în transfer intensiv de date. Utilizatorul care prelucrează, de exemplu, un text în foreground, aşteaptă cîteva secunde apariţia pe ecran a caracterelor tastate.

În concluzie: Desklink şi Kirschbaum Link nu reprezintă reţele adevărate, ele permiţînd doar accesul în comun la periferice (harddisk-uri, imprimante).

2. Lantastic Z - Reţea pentru 2 PC-uri, fără hardware suplimentar

Date tehnice:

- posturi de lucru
  nr. maxim: 2
- utilizare server ca post de lucru
  da
- necesar memorie pe server
  40 KByte
- hardware necesar
  PC/XT/AT/386 cu MS-DOS de la V.3.1
  interfaţă serială sau paralelă
- software de reţea admis
  orice soft de reţea compatibil IBM NetBios
- ofertant
  Microware, Düsseldorf

- preţ
  599 DM

Se recomandă atunci cînd, la un preţ convenabil, doriţi să utilizaţi în comun pe două calculatoare, baze de date sau periferice, avînd asigurată şi o protecţie corespunzătoare a accesului la date, cu viteze medii.

Dacă doriţi mai mult decît să copiaţi fişiere de la un calculator la altul, cum ar fi utilizarea unei baze de date în comun, nu mai sînt suficiente aşa numitele link-programs (ca Desklink sau Kirschbaum Link). Acestea permit accesul reciproc la periferice (harddisk, imprimantă), dar nu posedă mecanismele de securitate ale unui sistem de reţea adevărat.

Lantastic Z realizează o reţea adevărată cu două calculatoare prin intermediul interfeţei seriale sau paralele oferind toate funcţiile de reţea:

- ¤ - file şi recordlocking (interblocarea accesului simultan la acelaşi fişier sau articol).
- ¤ -acces protejat la drive-uri, directoare, fişiere
- ¤ -poştă electronică (comunicaţii prin teletransmisie)

Astfel Lantastic Z este complet compatibil cu aşanumitul standard Net-Bios. Aceasta înseamnă că produse speciale pentru reţele vor funcţiona şi sub Lantastic Z şi vor putea fi utilizate de ambele calculatoare.

Pachetul de software se livrează cu două cabluri, unul pentru interfaţă serială, altul pentru interfaţă paralelă. Viteza de transmisie este de 115 KBit/sec, pentru interfaţa serială şi de maximum 500 KBit/sec. pentru interfaţa paralelă.

Pentru cei care doresc în viitor să-şi extindă reţeaua la mai multe calculatoare, sau doresc o viteză de transfer mai mare, se oferă şi o versiune Lantastic cu hardware de reţea (plăci, cabluri) care permite interconectarea a pînă la 120 de posturi de lucru. Interfaţa utilizator şi setul de funcţiuni sînt identice la cele două versiuni.

Există şi posibilitatea conectării unui Laptop, prin intermediul interfeţei seriale, la server-ul unei reţele Lantastic.

Prin conectarea unui modem la server, Lantastic Z permite intrarea în reţea pe linie telefonică a oricărui PC care dispune de un modem şi soft Lantastic Z, binenţeles avînd şi drepturile de acces la reţea corespunzătoare (nume utilizator, parola, etc.). Astfel acesta poate utiliza de la distanţă resursele reţelei (baze de date, periferice), la viteze de transmisie de 1200/2400 bit/sec.

Suplimentar se oferă un produs numit "The Network Eye" (Ochiul reţelei), care este un program de supraveghere a tuturor staţiilor din reţea de pe o staţie oarecare. Acest program funcţionează pe orice reţea compatibilă cu standardul Net-Bios (ex. reţea Novell, reţea OS/2).

3. Reţea Kirschbaum

Date tehnice:

- posturi de lucru
  max. 255
- utilizare server cu post de lucru
  orice calculator poate fi server sau staţie
- necesar memorie
  22 KByte
- hardware necesar
  PC/XT/AT/386 cu MS-DOS de la V.3.01
  interfaţă serială sau paralelă
  placă de reţea ARCNET(opţional)
- software de reţea admis
  orice soft compatibil cu standardul IBM Net-Bios
- ofertant
  Kirschbaum Software Emmering
- preţ
  prevăzut 500 DM.

Se recomandă atunci cînd, la un preţ convenabil, doriţi să utilizaţi pe mai mult de două calculatoare în comun baze de date, periferice sau transferări simultane de date, fără a avea nevoie de o protecţie deosebită a accesului la date sau viteze ridicate de transfer.

Domeniul de utilizare recomandat: acolo unde doriţi să realizaţi o reţea adevărată cu 3-5 posturi de lucru, pentru bani puţini.

Reţeaua Kirschbaum este deocamdată singurul produs care permite realizarea unei reţele reale fără hardware suplimentar de maximum 255 calculatoare (teoretic).

Transferul de date se realizează prin intermediul interfeţelor seriale sau paralele.

Opţional există posibilitatea utilizării plăcilor de reţea ARCNET.

Topologia reţelei este foarte flexibilă. Astfel, pot fi interconectate de exemplu 3 calculatoare prin intermediul interfeţelor seriale iar alte două în continuare prin intermediul interfeţelor paralele. Se pot realiza şi topologii "stea" sau "inel", sau combinaţii ale acestora.

Soft-ul de reţea este complet compatibil cu standardul Net-Bios.

Nu există deosebiri între Server şi staţii: orice calculator din reţea are acces la perifericele (harddiskuri, imprimante) tuturor celorlalte calculatoare.

Dezavantaj: nu există funcţiuni de protecţie a accesului la date (parole, grupe de utilizatori, drepturi de acces). Totuşi, fiecare utilizator poate decide care periferice să fie utilizare numai local şi care în reţea.

Viteza de transfer este între 100-500 KBit/sec. şi depinde foarte mult de topologia reţelei. Pentru cei care doresc viteze ridicate de

transfer se recomandă, şi este posibilă, utilizarea plăcilor de reţea ARCNET (2,5 MBit/sec.)

La cumpărarea soft-ului de la Kirschbaum, utilizatorul primeşte o licenţă de reţea completă, ceea ce înseamnă că extinderea reţelei (ca număr de calculatoare) nu necesită plăţi suplimentare pentru soft.

Un domeniu de utilizare recomandat este de exemplu pentru acei programatori care doresc să dezvolte aplicaţii de reţea, fără să investească mulţi bani.

4. Fischernet Starterkit

Date tehnice:

- posturi de lucru
  maximum 4
- utilizare server cu post de lucru
  da
- necesar memorie pe server
  88 Kbyte plus 3 Kbyte pentru fiecare post de lucru suplimentar
- necesar memorie pe staţie de lucru
  70 Kbyte
- hardware necesar
  PC/XT/AT/386 cu MS-DOS începînd de la V 3.3
  plăci de reţea ARCNET sau de producător
- software de reţea admis
  orice soft compatibil cu standardul IBM Net-Bios
- ofertant
  Fischer Elektronik, Beilstein
- preţ
  Starterkit pentru 3 calculatoare, inclusiv cabluri, plăci ARCNET şi software: 1000 DM

Se recomandă atunci cînd doriţi să realizaţi o reţea ieftină cu 3 calculatoare, pentru a utiliza în comun baze de date, periferice şi puneţi accent atît pe securitatea accesului la date cît şi pe viteza de transfer.

Starterkit-ul oferit de firma Fischernet dispune de 3 plăci de reţea ARCNET (2,5 MB/sec.), trei cabluri coaxiale, confecţionate, de 5 m lungime, o doză de distribuţie, pachetul de software şi manualul de utilizare. Se pot livra şi alte tipuri de plăci de reţea:

- Toplink (1MB/sec.), mai ieftine
- Ethernet (10 MB/sec.), mai scumpe

Fischernet face deosebire între staţii de lucru şi server. Unul din calculatoare este configurat ca server, avînd conectate la el şi imprimantele de reţea. El poate fi utilizat totuşi în continuare şi ca post de lucru. Celelalte calculatoare au acces la discurile şi imprimantele server-ului.

Ca o problemă ar fi de luat în considerare necesarul de memorie pe server (88 Kbyte + min. 3 Kbyte). Dacă se utilizează şi ca post de lucru, nu mai rămîne spaţiu suficient pentru aplicaţii mai mari, cum ar fi dBase IV.

Pe lîngă versiunea de bază se oferă o serie întreagă de programe ajutătoare, care la alte tipuri de reţele sînt foarte scumpe.

Pentru scopuri de şcolarizare este oferită funcţia de supraveghere. Administratorul de reţea poate să-şi afişeze conţinutul ecranului oricărui post de lucru. Aceasta se poate face la oricare post de lucru.

Imprimanta de reţea poate fi conectată oriunde în reţea. Soft-ul de reţea este compatibil cu standardul Net-Bios.

Important: producătorul recomandă să nu folosiţi pe server MS-DOS versiunea 4.0 Aceasta poate conduce la anumite probleme, dacă se doreşte utilizarea unor harddisk-uri de capacitate mai mare, deoarece Fischernet nu permite decît maximum 4 drive-uri de reţea. Ori, pentru versiuni MS-DOS mai mici de 4.0, este permisă utilizarea unor partiţii de maximum 32 Mbyte, ceea ce limitează capacitatea maximă a spaţiului pe harddisk la 4 x 32 = 128 MByte.

Securitatea accesului la date este asigurată de Fischernet prin tot sistemul de parole, drepturi de acces utilizatori la directoare, fişiere, apartenenţă la grupe de utilizatori.

Deci dacă ştiţi că nu aveţi nevoie de mai mult de 4 PC-uri în reţea, vă recomandăm acest produs oferit de firma Fischer.

5. Lantastic

Date tehnice:

- posturi de lucru
  maximum 120
- utilizare server ca post de lucru
  da
- necesar memorie pe server
  40 Kbyte
- necesar memorie pe punct de lucru
  14 Kbyte
- hardware necesar
  PC/XT/AT/386 cu MS-DOS începînd cu V.3.1
  plăci de reţea diverse
- software de reţea admis
  orice soft compatibil Net-Bios
- producător
  ARTISOFT, S.U.A.
- ofertant
  Microware, Düsseldorf
- -preţ
  Starterkit cu 2 plăci Ethernet pe 16 biţi: 1975 DM
  Licenţă pentru 6 posturi de lucru, fără plăci de reţea: 1250 DM

Se recomandă atunci cînd doriţi realizarea unei reţele de maximum 120 calculatoare (teoretic) la viteză ridicată de transfer, şi puneţi accent pe economisirea memoriei ocupate cît şi pe protecţia datelor.

Reţeaua Lantastic a casei de software ARTISOFT s-a dovedit în teste a fi cea mai performantă, la un necesar de memorie foarte redus.

Nu este lipsit de interes să lăsaţi să funcţioneze pe server aplicaţii intensive, deoarece nu reduc foarte mult din viteza întregii reţele.

Se pot utiliza orice plăci de reţea compatibile cu NE 2000 ale firmei Novell. Firma Artisoft oferă plăci de reţea proprii pe 16 biţi compatibile Ethernet, însoţite de driver-ele corespunzătoare.

Dacă puneţi mare accent pe viteze de transfer, şi nu doriţi să utilizaţi server-ul şi ca post de lucru, puteţi instala un program LANCACHE, care accelerează accesul la date.

Se pot conecta pînă la trei imprimante pe server, la alegere pe interfeţele LPT1, LPT2, LPT3, COM1 sau COM2. Important: Se poate lista simultan la toate cele 3 imprimante.

Sistemul de securitate a datelor este foarte bine pus la punct. Pe lîngă protecţia prin parolă, Lantastic administrează drepturile de acces la reţea la toate nivelele: drepturi de acces pentru citire, scriere, ştergere de fişiere pot fi stabilite pentru orice utilizator şi orice fişier. Acelaşi lucru este valabil şi la nivel de directoare. De asemenea Lantastic ţine o evidenţă a tuturor cererilor de acces la date sau programe.

Toate funcţiunile de securitate sînt uşor de utilizat prin meniuri, cu toată complexitatea lor.

Artisoft oferă o serie de programe ajutătoare (care trebuie plătite suplimentar). Astfel, este oferit "The Network Eye" (ochiul reţelei), care permite unui utilizator să "comute" pe alt calculator din reţea şi să lucreze ca şi cînd ar fi la tastatura acelui calculator. Alt program permite utilizarea unui modem în comun de către toate posturile de lucru din reţea. Acesta poate fi conectat la oricare din calculatoare. Există şi un program care permite conectarea unui Laptop la o reţea Lantastic prin intermediul interfeţei seriale.

Versiunea de bază cuprinde şi un software de diagnoză, care conţine o funcţiune de dialog între doi utilizatori, cît şi o funcţiune de poştă electronică.

6. Invisible Ethernet

Date tehnice:

- posturi de lucru
  maximum 255
- utilizare server ca post de lucru
  da
- necesar memorie pe server
  3 - 100 Kbyte
- necesar memorie pe post de lucru
  3 - 70 Kbyte
- hardware necesar
  PC/XT/AT/386 cu MS-DOS începînd cu V.3.01
  plăci de reţea ale producătorului
- software de reţea admis
  orice soft compatibil cu standardul Net-Bios
- producător
  Invisible Software, S.U.A
- ofertant
  Ergos, Siegburg
- preţ
  pentru 1 calculator Software + placă Ethernet pe 16 biţi 990 DM

Se recomandă atunci cînd doriţi realizarea unei reţele performante, spaţiul de memorie ocupat fiind minim şi vă puteţi permite o cheltuială de 1000 DM per post de lucru.

Caracteristica deosebită a acestei reţele este necesarul foarte redus de memorie, software-ul utilizînd toate trucurile pentru a elibera cît mai mult spaţiul de memorie de lucru.

Astfel pe calculatoare care dispun de setul NEAT-chip (aşa numita memorie SHADOW-RAM) Invisible Ethernet se copiază aproape în întregime în spaţiile de memorie nefolosite, peste limita celor 640 Kbyte. Memoria SHADOW-RAM ocupă cei 384 Kbyte, deasupra limitei de 640 Kbyte a sistemului de operare MS-DOS, care sînt rezervaţi şi pentru BIOS, adaptor video, alte extensii de sistem. O parte din acest spaţiu, şi anume cel care este în paralel cu BIOS-ul sistemului şi al adaptorului video este utilizat pentru accelerarea acceselor la memorie. Spaţiul rămas disponibil este ocupat de Invisible Ethernet, astfel încît nu mai necesită din memoria de lucru decît maximum 35 Kbyte.

Versiunea cea mai nouă de Invisible Ethernet funcţionează şi cu Windows 3.0 fără probleme. Administrarea memoriei a fost realizată de către producător astfel încît partea de reţea să nu ocupe decît 2 - 4 Kbyte de memorie, astfel încît Windows 3.0 funcţionează stabil şi în modul extins 80386.

Se pot conecta la server maximum 2 imprimante

Dezavantaj: Nu se poate face listare decît la una din imprimante la un moment dat.

Se pot utiliza numai plăci de reţea ale producătorului: Invisible Ethernet. Software-ul de reţea este compatibil cu standardul Net-Bios. Reţeaua funcţionează atît sub MS-DOS cît şi sub DR-DOS 5.0, sau combinat.

Sistemul de securitate a datelor cuprinde atît parole cît şi drepturi de acces la diferite nivele. Nu este prevăzută securitatea fişierelor singulare.

7. Novell ELS I şi ELS II

Date tehnice:

- posturi de lucru
  ELS I: 4 ELS II: 8
- utilizare server ca post de lucru
  da
- necesar memorie pe server
  100 Kbyte

- necesar memorie pe staţie de lucru
  70 Kbyte
- hardware necesar
  PC/XT/AT/386 (server numai AT sau 386) cu MS-DOS începînd cu V.3.0
  plăci de reţea de diferite standarde şi producători
- software de reţea admis
  orice soft compatibil cu standardul Net-Bios sau Novell
- preţ
  ELS I 2177 DM (pentru 4 posturi de lucru)
  ELS II 5187 DM (pentru 8 posturi de lucru)

Se recomandă atunci cînd doriţi să utilizaţi software sub standardul Novell pe maximum 8 posturi de lucru, punînd un accent deosebit pe securitatea şi protecţia datelor, eventual pentru posibilitatea de a conecta şi calculatoare MACINTOSH, şi dispuneţi de aproximativ 5000 DM

Novell reprezintă încă liderul de necontestat al pieţii reţelelor de calculatoare. În special pentru reţele mai mari sînt oferite multe pachete de software, concepute special pentru astfel de sisteme.

În timp ce alte sisteme de reţele reprezintă de fapt doar nişte programe suplimentare care extind MS-DOS la funcţiuni de reţea, Novell Netware este el însuşi un sistem de operare. Astfel Novell utilizează un format propriu de înregistrare a datelor pe harddisk. Acest lucru oferă pe de o parte o siguranţă suplimentară. Cu funcţia "Read-after-Write", de exemplu, se verifică la orice scriere, dacă datele au fost înregistrate fără eroare pe disc. Dacă se detectează o eroare, blocul este marcat ca defect şi se reia operaţia de scriere în alt loc pe disc. Toate aceste operaţii se realizează complet automat.

Sistemul de operare de reţea ţine o evidenţă foarte exactă a tuturor acceselor asupra fişierelor: nu este memorată numai data modificării unui fişier ci şi de către care utilizator a fost ultima oară citit sau scris. Aceste informaţii sînt foarte utile în special la protecţia datelor şi la arhivare.

Un dezavantaj faţă de alte sisteme de reţele este durata mai mare de instalare a software-ului pe server.

Siguranţa în funcţionare şi securitatea datelor sînt pe primul plan la sistemele Novell. Software-ul de reţea sprijină în acest sens şi surse de alimentare neîntreruptibile (UPS) care asigură funcţionarea server-ului în caz de cădere a tensiunii de alimentare.

Sistemul posedă şi posibilitatea detectării intruşilor din reţea, adică a celor care nu cunosc o parolă şi încearcă să intre în reţea.

Ca server se poate utiliza şi un AT 80286, care poate fi folosit şi ca staţie de lucru, reducîndu-se considerabil viteza de transfer în reţea.

Versiunile reduse de Novell, ELS I şi ELS II nu sînt mai complicate la instalare faţă de alte sisteme, dar deoarece cuprind peste 10 dischete, necesită un timp de instalare mai îndelungat. Dar munca propriu-zisă abia acum începe: trebuie stabilite drepturile de acces, instalate imprimante, programe de utilizator, etc.

Novell Netware este cel mai scump din sistemele prezentate. În majoritatea cazurilor utilizatorul nu foloseşte la întreaga capacitate funcţiunile extinse ale sistemului şi posibilităţile suplimentare de securitate.

Instalarea unei reţele Novell Netware se justifică numai dacă toate aceste funcţiuni suplimentare sînt într-adevăr folosite.

8. Reţea XIRCOM

Date tehnice:

- hardware necesar
  PC/XT/AT sau LAPTOP
  interfaţă paralelă
  adaptor XIRCOM
- software necesar
  MS-DOS începind cu V.3.3, soft de reţea
- ofertant
  Computer 2000, München
- preţ

Token Ring	2.700 DM
Ethernet	2.200 DM

Se recomandă atunci cînd doriţi să aveţi acces rapid la o reţea cu un LAPTOP, fără să fie necesară montarea unei plăci suplimentare în calculator.

Adaptorul de reţea XIRCOM, produs de firma americană Xircom, este de dimensiuni foarte reduse ("Pocket-Adapter"), şi se conectează direct la interfaţa paralelă.Sînt oferite adaptoare pentru reţele de tip Ethernet, Thin-Ethernet, Token Ring, şi Arcnet.

Alimentarea adaptorului se face printr-un cablu livrat de firmă. Instalarea software-ului este foarte simplă: se copiază 1-3 fişiere pe o dischetă, în funcţie de tipul reţelei (adaptorului). Apoi LAPTOP-ul are acces în reţea ca o nouă staţie de lucru.

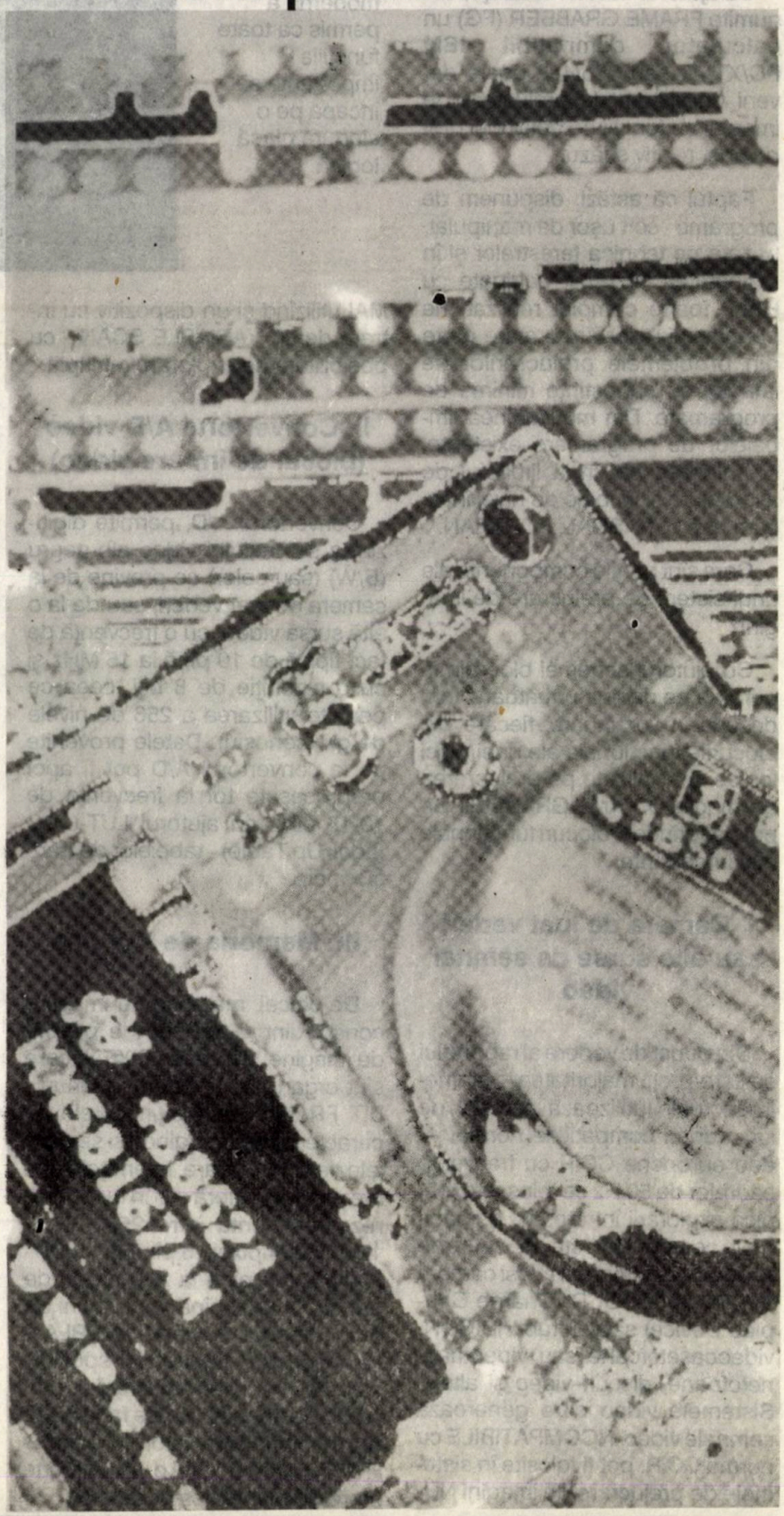
Odată cu adaptorul se livrează şi o serie de programe driver pentru software-ul Novell Netware: 3Com 3+, IBM-PC Lan, NetBios, TCP/IP, MS-LAN Manager, Sun-PC-NFS, Artisoft Lantastic, Corvus-PC-NOS, Banyan Vines.

Viteza de transfer a datelor (100-500 kb/sec) este destul de redusă, în comparaţie cu cea a plăcilor de reţea.(2-10 Mb/sec), dar nu întotdeauna aceasta reprezintă un factor decisiv în alegerea unei soluţii de interconectare.

(E. P.)

Bazele prelucrării de imagini pe calculatoare personale

Prelucrarea numerică a imaginilor (Digital Image Processing - DIP), pe PC-uri, reprezintă o nouă "tehnologie", care s-a răspîndit foarte repede, în domenii ca tehnica măsurărilor, a reglării automate şi a comenzilor; cu ajutorul unei camere de luat vederi, cu funcţie de senzor, şi a unui PC se pot rezolva probleme ce pînă acum erau posibile de rezolvat doar pe baza complexei interacţiuni ochi-creier uman.



Aplicaţii tipice ale prelucrării de imagini sînt întîlnite azi aproape în fiecare laborator, în cercetare sau în industrie. Cele mai importante dintre acestea sînt:

- -analiza de imagini
- -evaluarea imaginilor röntgen
- -comanda roboţilor
- -controlul automat al fabricaţiei
- -controlul calităţii produselor
- -recunoaşterea formelor
- -evaluarea imaginilor termice
- -grafică pe calculator
- -analiza mişcărilor
- -instalaţii de pază şi securitate
- -preluarea şi evaluarea imaginilor din satelit
- -tehnica DTP - DeskTop Publishing
- -preluarea şi interpretarea imaginilor microscopice

Pînă în prezent, sistemele de manipulare a imaginilor au fost foarte costisitoare. Acestea, pe lîngă un calculator conţineau încă multe alte componente, alcătuind astfel un sistem foarte complex, greu de programat şi care se preta doar la aplicaţii speciale.

Odată cu integrarea unor componente electronice, cum ar fi convertoarele analog-numerice (A/D) şi numeric-analogice (D/A) VIDEO, disponibilitatea unei memorii tot mai mari per cip etc., au permis realizarea unui sistem "universal"

de prelucrare de imagini pe o singură placă logică PC-BUS compatibilă.

Cu ajutorul acestei interfeţe, denumite FRAME GRABBER (FG) un calculator compatibil IBM PC/XT/AT sau 386/486 poate deveni un sistem de prelucrare de imagini cu funcţii complexe şi preţ de cost relativ scăzut.

Faptul că astăzi, dispunem de programe - soft uşor de manipulat, bazate pe tehnica ferestrelor şi în majoritatea cazurilor echipate cu HELP foarte complet realizat, ne permite să rezolvăm o mare parte din problemele prelucrărilor de imagini cu cunoştinţe minime de programare. Din multitudinea limbajelor de programare existente, se folosesc pe lîngă limbajul de asamblare şi limbaje de nivel înalt, cum ar fi C, PASCAL, FORTRAN.

Care sînt părţile componente ale unui sistem de prelucrare de imagini?

Cu ajutorul schemei bloc din figura de pe pagina următoare vom descrie în continuare, fiecare din blocurile funcţionale ale unei plăci logice specializate pe prelucrarea de imagini - FRAME GRABBER. Se pot desprinde 6 blocuri funcţionale mai importante:

I. Camera de luat vederi sau alte surse de semnal video

Din punct de vedere al raportului calitate/preţ, majoritatea sistemelor actuale utilizează camere de luat vederi compatibile normei video europene CCIR cu frecvenţa cadrelor de 50 Hz ce foloseşte tehnica explorării întreţesute a imaginii. În ceea ce priveşte alte surse de semnal video, putem folosi camere de luat vederi CCD (Charge Coupled Device) sau cu tub VIDICON, videocasetofoane sau videomagnetofoane, discuri video şi altele. Sistemele video care generează semnale video INCOMPATIBILE cu norma CCIR, pot fi folosite în sistemele de prelucrare de imagini NUMAI utilizînd şi un dispozitiv cu intrare de tip VARIABLE SCAN - cu posibilitate de explorare multiplă.

Tehnologia modernă a permis ca toate funcţiile importante să încapă pe o singură placă logică

II. Convertorul A/D video (blocul de intrare video)

Convertorul A/D, permite digitizarea semnalului video alb-negru (B/W) (sau color) ce provine de la camera de luat vederi, sau de la o altă sursă video, cu o frecvenţă de tact tipică de 10 pînă la 15 MHz şi cu o rezoluţie de 8 biţi, ceea ce permite utilizarea a 256 de nivele de gri (cenuşiu). Datele provenite de la convertorul A/D pot fi apoi preprocesate tot la frecvenţa de 10-15 MHz cu ajutorul LUT-urilor (Look Up Table) - tabelelor de transformare.

III. Memoria de imagine

De obicei, memoria de imagine constă dintr-o matrice de puncte de imagine, denumite PIXELI, care sînt organizate în diferite planuri - BIT FRAME, ce conţin datele ce caracterizează imaginea şi semnalele de sincronizare. Interfeţele oferite pe piaţă, diferă aici în funcţie de rezoluţia dorită între 256*256 şi 1280*1024 pixeli, 6 pînă la 12 BIT-FRAME (64 pînă la 256 nivele de gri) şi opţional pînă la 4 pagini de memorie suplimentară - OVERLAY. La majoritatea interfeţelor specializate pe prelucrări de imagini, memoria de imagine poate face parte din memoria calculatorului gazdă - HOST, oferind astfel o viteză foarte mare de manipulare a datelor.

IV. Blocul de procesare

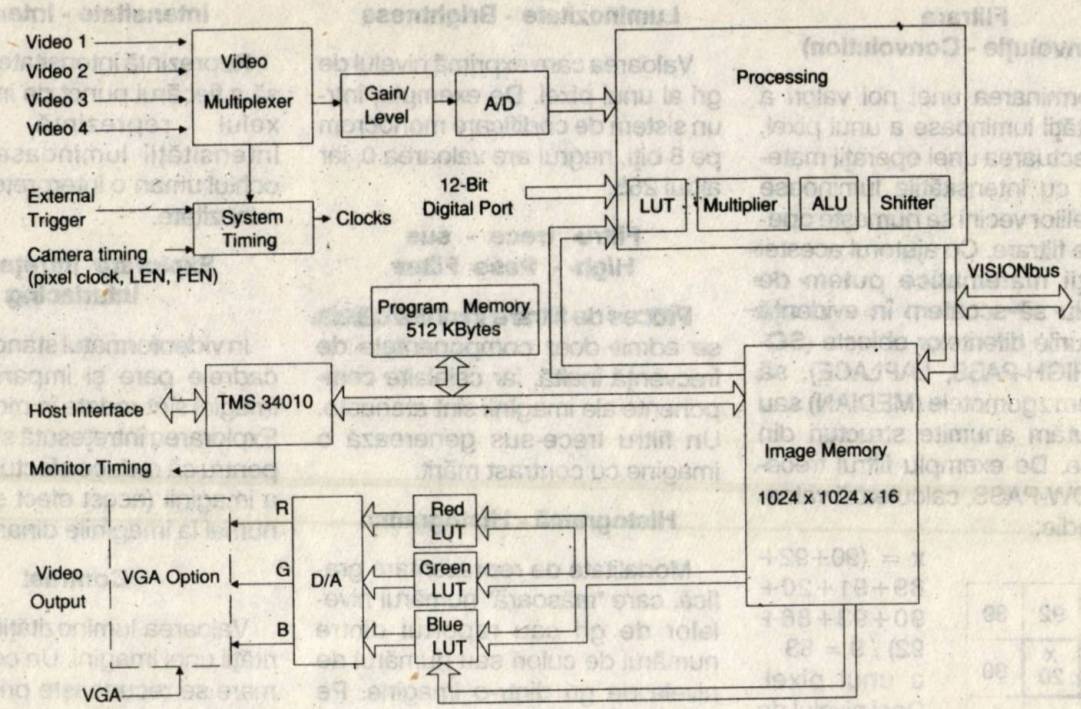
Multe dintre sistemele DIP dispun de un bloc de procesare ce poate efectua multe operaţii matematice în timp real (40 ms), iar altele, aproape în timp real. Operaţii tipice de timp real sînt operaţiile de combinare a două imagini ca de exemplu scăderea, însumarea, sau multiplicarea acestora.

V. Convertorul D/A (blocul de ieşire video)

Convertorul D/A preia funcţia de reconstrucţie a informaţiei video numerice într-un semnal analogic şi generarea semnalelor de sincronizare (linii şi cadre). Toate sistemele DIP conţin cîte 3 convertoare D/A, de 8 biţi fiecare, cu ajutorul cărora se poate obţine o imagine PSEUDO-COLOR din informaţia alb-negru. LUT-uri de 8 biţi montate la ieşire, permit obţinerea de imagini color cu 256 de culori simultane dintr-o paletă de 16 milioane de culori posibile. Interfeţele din generaţia avansată - ADVANCED FRAME GRABBER (AFG), conţin în plus şi un procesor grafic care realizează funcţia de afişare a imaginii pe un monitor grafic.

VI. Compatibilitatea cu magistrala calculatorului gazdă - HOST

Pe piaţă există interfeţe FG aproape pentru orice tip de calculator existent. Astfel, există FG-uri



Schema bloc a unei interfeţe de procesare de imagini (de ex. AFG a firmei Imaging Technology)

compatibile cu BUS-ul PC/XT/AT, PS/2 (MCA - Micro Channel Architecture), Q-BUS, VME BUS. Unele FG-uri conţin în plus un conector de interfaţă care permite de exemplu ca un FG compatibil VME BUS să funcţioneze tot atît de bine şi pe un PC/XT/AT, PS/2, HP 9000, APOLLO sau SUN.

VII. Software

Majoritatea sistemelor DIP se livrează începînd cu programe de test şi diagnoză, drivere soft scrise în diferite limbaje de nivel înalt şi terminînd cu pachete soft complexe bazate pe meniuri, ce conţin o paletă largă de funcţii privind analiza şi prelucrarea de imagini, numite INTERPRETOARE.

Prelucrări de imagini minilexicon de termeni

Convertor A/D

Hardware, care realizează conversia unui semnal video analogic în informaţie binară. Acest proces se numeşte DIGITIZARE. În prelucrarea de imagini, digitizarea unui semnal analogic se face pe 8 biţi = 256 nivele de gri.

Unitate aritmetică şi logică (ALU - Arithmetic Logic Unit)

Hardware, ce realizează operaţii aritmetice ca de exemplu adunare, scădere sau operaţii logice ca ŞI, SAU. COMBINAREA este o operaţie ce se execută între imaginea de referinţă şi imaginea achiziţionată în timp real numită imaginea vie - LIVE IMAGE.

Rezoluţie - Resolution

Numărul de nivele de gri utilizate pentru redarea luminozităţii unui pixel. De exemplu o rezoluţie de 8 biţi = 256 nivele de gri, iar o rezoluţie de 6 biţi = 64 nivele de gri.

Frecvenţa cadrelor Frame Frequency

Numărul de cadre redate într-o secundă; în videoformatul european CCIR = 25 cadre/secundă, 50 semicadre/secundă; în cazul normei RS 170 USA standard = 30 cadre/secundă, 60 semicadre/secundă.

Raport de aspect - Aspect Ratio

Raportul dintre lăţimea şi înălţimea imaginii. Raportul standard este de 4:3 la camerele video realizate cu tuburi, CCD şi la monitoarele TV.

Crominanţa - Crominance

Deosebire metrică (lungime de undă dominantă şi saturaţie) între culori şi legătura acestora cu lumina "albă", păstrînd luminozitatea constantă.

Semnal video complex Composite Video Signal

Semnal ce rezultă din însumarea uneia din componentele semnalului color, de exemplu VERDE - G, cu semnalele de sincronizare necesare unui monitor TV.

Imagine numerică Digital Image

Imagine compusă din pixeli (puncte discrete de imagine) cu diferite luminozităţi, organizate sub forma unei matrici, puncte de exemplu de 512*512 sau 768*512.

Filtrare (convoluţie - Convolution)

Determinarea unei noi valori a intensităţii luminoase a unui pixel, prin efectuarea unei operaţii matematice cu intensităţile luminoase ale pixelilor vecini se numeşte operaţie de filtrare. Cu ajutorul acestei operaţii matematice putem de exemplu să scoatem în evidenţă contururile diferitelor obiecte (SOBEL, HIGH-PASS, LAPLACE), să reducem zgomotele (MEDIAN) sau să căutăm anumite structuri din imagine. De exemplu filtrul trece-jos - LOW-PASS, calculează valoarea medie:

90	92	89
91	x 20	90
93	88	92

x = (90+92+89+91+20+90+93+88+92) / 9 = 83 a unui pixel. Deci nivelul de gri mediu este de 83.

Cu ajutorul acestui operator, putem reduce frecvenţele înalte din imagine (variaţiile bruşte ale nivelelor de gri).

Cadru - Frame

Suma tuturor punctelor explorate care generează o imagine într-un aparat de reprezentare a imaginii. Într-un sistem de explorare întreţesută (2 semicadre) un cadru este intervalul de timp dintre începutul primului semicadru şi sfîrşitul celui de-al doilea semicadru.

Analiza în domeniul frecvenţă - Frequency Analysis

Modalitate de măsurare a componentelor din spectrul de frecvenţă, de-a lungul unui caracter selectat din imagine sau din interiorul unei arii de interes (Area Of Interest - AOI).

Nivel de gri - Grey Level

Variaţia în intensitate a luminii "albe" de la negru la alb. Nivelele de gri pot fi reprezentate într-un sistem de codificare monocrom pe 8 biţi între valorile 0 şi 255 (0 = NEGRU , 255 = ALB).

Luminozitate - Brightness

Valoarea care exprimă nivelul de gri al unui pixel. De exemplu, într-un sistem de codificare monocrom pe 8 biţi, negrul are valoarea 0, iar albul 255.

Filtru trece - sus High - Pass Filter

Proces de filtrare în urma căruia se admit doar componentele de frecvenţă înaltă, iar celelalte componente ale imaginii sînt atenuate. Un filtru trece-sus generează o imagine cu contrast mărit.

Histogramă - Histogramm

Modalitate de reprezentare grafică, care "măsoară" numărul nivelelor de gri sau raportul dintre numărul de culori sau numărul de nivele de gri dintr-o imagine. Pe axa orizontală se reprezintă nivelele de gri, iar pe cea verticală, numărul de pixeli ce au un acelaşi nivel de gri. Histograma ne dă o informaţie despre contrastul imaginii şi ajută astfel utilizatorul să se decidă asupra folosirii unui filtru (operator) de mărire a contrastului. Dacă prin rărirea liniilor unei histograme obţinem o mărire a contrastului, atunci se obţine o mai bună reprezentare a imaginii dar nu se obţine o îmbunătăţire a semnalului video. Mărirea contrastului, creşterea nivelului şi a offsetului duc la obţinerea unei variaţii în histogramă.

Intensitate - Intensity

Reprezintă intensitatea luminoasă a fiecărui punct de imagine. Pixelul reprezintă valoarea intensităţii luminoase pe care ochiul uman o interpretează ca luminozitate.

Explorare întreţesută Interlacing

În videoformatul standard, semicadrele pare şi impare ale unei imagini sînt redate în mod alternat. Explorarea întreţesută se foloseşte pentru că reduce efectul de pîlpîire a imaginii (acest efect se observă numai la imaginile dinamice).

Contrast

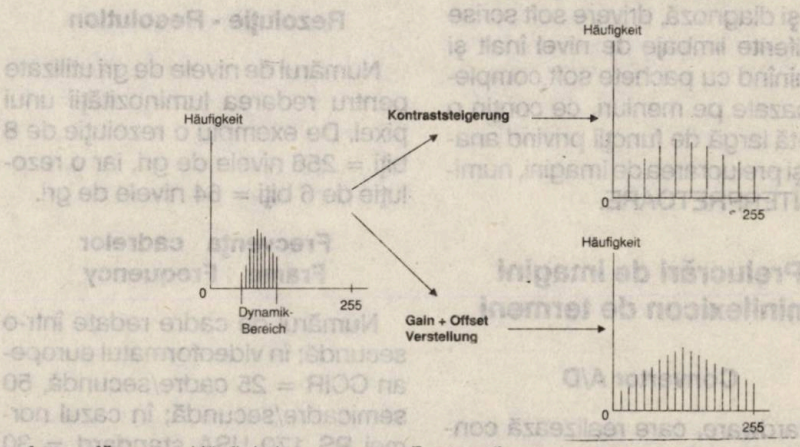
Valoarea luminozităţii sau a clarităţii unei imagini. Un contrast mai mare se recunoaşte prin faptul că imaginea conţine mai multe puncte negre şi albe. O imagine ce are un contrast mediu se caracterizează printr-un spectru gri, iar o imagine cu contrast redus are doar o slabă contribuţie în domeniul valorilor gri.

Îmbunătăţirea contrastului - Contrast Enhancement

Operaţie cu care contrastul unei imagini variază în sensul obţinerii unei imagini mai clare.

Tabelă de transformare - Look Up Table (LUT)

Funcţie hardware de bază, care, azi nu poate lipsi dintr-un sistem



Legendă:

Häufigkeit	= Frecvenţă
Kontraststeigerung	= Variaţia contrastului
Gain + Offset Verstellung	= Schimbarea în cîştig

Sintaxă: Parametri folosiţi ca atare sînt scrişi cu majuscule. Parametri scrişi cu litere mici ţin locul unei valori cu care trebuie înlocuiţi în momentul lansării comenzii (un nume de fişier de exemplu). <Opţiune> este un parametru opţional, iar bara verticală ("|") trebuie interpretată ca "sau".

Deoarece în aproape toate cazurile parametrul /H are ca efect afişarea help-textului, acest parametru nu a mai fost explicitat decît în cazul că are altă semnificaţie.

APPEND </H> <u:cale <;<u:>cale<...>>>
</X:ON|OFF> </E:ON|OFF>
</PATH:ON|OFF>

u:cale — specificator de unitate şi cale
<;<u:>cale<...>> — se pot specifica mai multe căi
</X:ON|OFF> — se (nu se) permit funcţiile extinse de căutare (.COM, .EXE, .BAT)
</E:ON|OFF> — păstrează lista APPEND în Environment (se poate vedea cu SET)
</PATH:ON|OFF> — APPEND se (nu se) utilizează şi pentru fişierele negăsite cu cale specificată sau respectiv şterge lista APPEND
nici un parametru — afişează lista APPEND

ASSIGN u1: = u2: <u1:=u2:><...> </A>

u1: = u2: — asignare de unitate disc; u1 = unitatea care se redirectează; u2: = unitatea pe care se face redirectarea
<u1:=u2:><...> — se pot face mai multe asignări deodată
</A> — afişează tabelul cu toate asignările în uz
nici un parametru — şterge lista de asignări

ATTRIB </H> <+|-A> <+|-H> <+|-R> <+|-S>
<@> <numefişier><...></P></S>

<+|-A> — setează/resetează bitul de arhivare
<+|-H> -II- hidden ("ascuns")
<+|-R> -II- read-only
<+|-S> -II- fişier-sistem
<@> <numefisier> <...> — (listă de) nume-fişier(e) cu * sau ?;
</P> — paginare la afişaj
</S> — se ţine seama şi de toate subdirectoarele

BACKUP </H> u_sursă:<numefişier> <u_destinaţie:> </S> </F> </M> </D:data>
</T:ora> </A> </L:<cale>logfile>

u_sursă:<numefişier> — de unde se salvează
<u_destinaţie:> — unitate pe care se salvează
</S> — se ţine seama şi de toate subdirectoarele
</F> — unitatea destinaţie se şi formatează
</M> — se salvează toate fişierele care au suferit modificări de la ultimul BACKUP
</D:data> -II- după această dată
</T:ora> -II- după această oră
</A> — fişierele ce se salvează se adaugă celor deja salvate
</L:<cale>logfile> — la salvare, se crează şi un fişier "proces-verbal al salvării"

BREAK <ON|OFF>

<ON|OFF> — ia / nu ia în seamă Ctrl-Break

DR DOS 5.0
Memento

u: — unitatea pe cre se află discheta de formatat
</S> — la sfîrşitul formatării, se vor trece şi fişierele sistem
</V> — FORMAT va cere un nume pentru dischetă
</1> — se formatează pe o singură faţă (320/360 kbyte)
</4> — se formatează dischete de 360 kbyte
</8> — se lucrează numai pe 8 sectoare
</B> — se lucrează numai pe 8 sectoare şi se rezervă loc pentru sistem
</T:xx> — numărul de piste (40,80)
</X:x> numărul de sectore (8,9)
</F:xxxx> — capacitatea dischetei (160, 180, 320, 360, 720, 1200, 1440)

GRAFTABL </H> <codepage> </STATUS>

<codepage> — 437|865|860|863|850 numere posibile pentru pagina de cod
</STATUS> — afişează numărul paginii de cod în uz

GRAPHICS </H> </COLOR> </R>

</COLOR> — permite imprimarea color pe imprimante IBM
</R> — inversează alb cu negru în imagine

HILOAD numefişier

numefişier — fişierul ce trebuie încărcat

JOIN </H> <u1:u2:cale> <u2:cale /D>

<u1:u2:cale> — unitatea ce urmează a fi substituită şi calea la care se face legătura
<u2:cale /D> — taie legătura înspre calea specificată
nici un parametru — afişează lista JOIN

KEYB </H> xx<+|-><,codetable>

xx — codul ţării pentru tastatură (US - S.U.A.)
<+|-> — dacă tastatura nu a fost recunoscută automat, se face cu "+" specificarea unei tastaturi de 101/102/104 taste şi cu "-" pentru 83/84 taste
<,codetable> — tabela de coduri (vezi şi GRAFTABL)

LABEL </H> <u:|u:nume>

<u:|u:nume> — dă un nume discului de pe unitatea u:

MEM </H> </B> </D> </S> </P> </M> </A>

</B> — afişează zonele ocupate de DRDOS
</D> — afişează numele şi poziţiile tuturor driverelor de aparate
</S> — afişează înlănţuirea tampoanelor de harddisk
</P> — paginare la afişare
</M> — arată grafic întreaga ocupare a memoriei
</A> — afişează toate informaţiile

MKDIR|MD <u:>nume_director

<u:>nume_director — directorul ce urmează a fi creat

MODE </H>

MODE mod_de reprezentare <,rînduri>

mod_de reprezentare — CO40, CO80, BW40, BW80, 40, 80, MONO
<,rînduri> — 25, 43, 50

MODE <mod_de reprezentare> ,m <,T>

<mod_de reprezentare> — CO40, CO80, BW40, BW80, 40, 80, MONO
,m — sensul (R=dreapta, L=stînga)
<,T> — afişează un ecran-mostră

MODE CON<:> <LINES=rînduri> <COLS=coloane> <RATE=r,DELAY=d>

<LINES=rînduri> — 25, 43, 50

DR DOS 5.0
Memento

Cache </H> </S=nnnn> </X> </E>
- </S=nnnn> mărimea tamponului în kilobytes
- </X> tamponul să fie în Expanded Memory
- </E> tamponul să fie în Extended Memory

CHCP <codepage>
- <codepage> comută pagina de coduri (437 | 850 | 860 | 863 | 865)
- nici un parametru afişează pagina de cod în uz

CHDIR|CD <u:>cale
- <u:>cale calea indicată devine calea curentă
- nici un parametru afişează calea curentă

CHKDSK </H> <u:><numefişier> </A> </B> </C> </D> </F> </L> </M> </P> </R> </S> </V>
- <u:> unitatea de verificat
- <numefişier> fişierul (fişierele) de verificat
- </A> info relative la mărimea şi ocuparea RAM
- </B> marchează blocurile defectuoase
- </C> afişează numerele de blocuri ale fişierelor defectuoase
- </D> caută subdirectoare şterse
- </F> se pot face corecturi pe discul verificat
- </L> încearcă să refacă FAT-uri defecte
- </M> elimină blocurile defectuoase
- </P> afişează blocurile ce revin directoarelor
- </R> reface directorul rădăcină
- </S> afişează suma dimensiunilor reale ale fişierelor
- </V> afişează informaţii suplimentare în timpul verificării

CLS

şterge ecranul

COMMAND <u:><cale><echipament de intrare&ieşire> </E:nnnnn> </P> </C> <comandă>
- <u:><cale> calea pe care se va căuta interpretorul de comenzi ce urmează a se încărca
- <echipament de intrare&ieşire> vezi CTTY
- </E:nnnnn> mărimea Environment-ului, în bytes
- </P> interpretorul de comenzi rămîne în RAM
- </C><comandă> comandă de executat. /C permite retur automat

COMP </H> <numefişier1> <numefişier2> </A> </M:xxx>
- <numefişier1> primul fişier
- <numefişier2> şi cel cu care se compară
- </A> deosebirile se afişează în ASCII
- </M:xxx> numărul de diferenţe admise (0 = nici una)

COPY sursa1 <+sursa2....> <destinaţie> </A> </B> </V> </S> </C> </Z>
- sursa1 (prima) sursă de unde se copiază; poate fi echipament sau fişier
- <+sursa2....> sursă(surse) suplimentară; poate fi doar fişier
- <destinaţie> unde se copiază; poate fi echipament sau fişier
- </A> sursa e fişier ASCII; CTRZ-Z = sfîrşit de fişier
- </B> sursa e fişier binar. CTRL-Z **nu** înseamnă sfîrşit de fişier

DR DOS 5.0
Memento

- <COLS=coloane> 40, 80
- <RATE=r,DELAY=d> r = 1..33, d = 1..5

MODE LPTx<:>=COMy
- LPTx<:> adaptorul paralel (x = 1,2,3,4) care se redirectează spre
- COMy adaptorul serial (y = 1,2,3,4)

MODE LPTx<:> <<mx>, <my>><,p>
- LPTx<:> adaptorul paralel (cu numărul x = 1,2,3,4)
- <<mx>,<my>> mx = caractere pe rînd, my = caractere pe inch
- <,p> imprimare fără limită de timp

MODE COMx<:> baudrate <,<paritate>,<biţi_de_date>,<biţi_de_stop>,<P>>
- COMx<:> adaptorul serial (x = 1,2,3,4)
- baudrate viteza de transmisie dorită (cele două cifre cele mai semnificative)
- <paritate> verificarea parităţii (O = impară, E = pară, N = indiferent)
- <biţi_de_date>,<biţi_de_stop> 8|7|6|5, 1|1.5|2
- <,P> transmisie fără limită de timp

MORE
- nici un parametru exemple de utilizare: sursă_date | MORE <| program_filtru> sursă_date | MORE > destinaţie_date

NLSFUNC </H> <numefişier>
- <numefişier> fişier cu informaţii relative la ţară (de ex. COUNTRY.SYS)

PASSWORD </H> <@><numefişier><...> </R|/W|/D|/G|/P<:parolă>><N> </NP> </NG> </S>
- <@><numefişier><...> (listă de) fişiere specificate cu * şi ?
- </R:parolă> parola necesară pentru citirea, copierea, scrierea, ştergerea, redenumirea sau modificarea atributelor fişierelor specificate
- </W:parolă> parola fără care fişierele pot fi numai citite
- </D:parolă> parola care apără fişierele specificate de ştergere sau redenumire; modificări de atribute sînt permise
- </G:parolă> parola globală standard
- </P:parolă> protecţie la subdirectoare
- </N> suspendă orice protecţie prin parole; parametrul poate fi folosit împreună cu /G sau /P
- </NP> anulează parola globală
- </NG> anulează protecţia prin parolă relativ la un director
- </S> se ţine seama şi de toate subdirectoarele

PATH <u:>cale <;<u:>cale><...>>
- <u:>cale unde se caută fişierele lansate ca şi comenzi (.COM, .EXE, .BAT)
- <;<u:>cale><...>> se pot specifica mai multe căi, directoare, subdirectoare

PRINT </H> </B:dimensiune> </U:tactul de aşteptare> <M:tactul de imprimare> </S:cuanta de timp> </Q:coada de aşteptare> </T> <<numefişier> </L:echipament> </C> </P>> <...>
- <numefişier> fişierul (fişierele) de imprimat
- </L:echipament> echipamentul de ieşire (PRN, LPTx:, AUX...)

</V> cu verificare
</S> se copiază şi fişierele sistem
</C> copiere cu consultarea utilizatorului
</Z> şterge bitul superior al fiecărui byte copiat

CTTY echipament_de_intrare&ieşire
echipament_de_intrare&ieşire poate fi CON, COMx, AUX

CURSOR </H> </Sxx> </C> </OFF>
</Sxx> intervalul de clipire, în 1/18 secunde
</C> evită "zăpada" CGA
</OFF> comută înapoi pe cursor hard

DATE <zz-ll-aa>
<zz-ll-aa> setează data curentă (zi, lună an)
nici un parametru afişează data curentă

DEL numefişier </C> </S>
numefişier fişier(e) de şters
</C> cu consultarea utilizatorului
</S> şterge şi fişierele sistem

DELQ numefişier </S>
echivalent cu DEL /C

DIR <numefişier> </W|/L> </D|/S|/A> </P|/N> </R> </C>
<numefişier> nume fişier despre care se cer informaţii
</W|/L> format lat "wide") sau lung ("long") pentru afişaj
</D|/S|/A> selecţie fişiere: /D: fişiere normale, /S fişiere sistem, /A: toate fişierele
</P|/N> afişaj paginat sau nu
</R> preia noile valori ale parametrilor ca standard
</C> preia valorile specificate pentru parametri ca standard pentru următorul apel

DISKCOMP </H> <u:<u:>> </1> </8> </V>
<u:<u:>> unităţile ce se compară
</1> se verifică numai prima faţă a dischetei
</8> se verifică numai primele 8 sectoare
</V> verifică dacă discheta poate fi citită în întregime

DISKCOPY </H> <sursa> <destinaţie> </1>
<sursa> unitatea sursă
<destinaţie> unitatea destinaţie
</1> se copiază numai prima faţă

EDITOR </H> <numefişier>
<numefişier> fişierul de editat

ERAQ numefişier </S>
echivalent cu ERA /C

ERASE|ERA numefişier </C> </S>
numefişier fişierul (fişierele) de şters
</C> se cere acordul utilizatorului
</S> se şterg şi fişiere sistem

EXE2BIN </H> numefişier1<.tip> <numefişier2<.tip>> </Sxxxx>
numefişier1<.tip> fişierul de convertit (.EXE)
<numefişier2<.tip>> fişierul în care se face conversia (.COM)
</Sxxxx> valoare hexa pentru cazul corecturilor de

DR DOS 5.0
Memento

</B:dimensiune> dimensiunea buffer-ului, (bytes)
</U:tactul de aşteptare> numărul tactelor de aşteptare (implicit 1)
<M:tactul de imprimare> numărul tactelor de imprimare pe care are voie PRINT să le folosească înainte de a ceda UC pentru celelalte activităţi (implicit 2)
</S:cuanta de timp> numărul de tacte disponibile pentru alte activităţi cînd PRINT e activ
</Q:coada de aşteptare> cîte fişiere pot intra în coada de aşteptare
</T> goleşte coada de aşteptare
</C> şterge fişierul anterior şi toate cele următoare din coada de aşteptare (pînă la un /P)
</P> adaugă fişierul anterior şi toate cele următoare la coada de aşteptare (pînă la un /C)

PROMPT <promptstring>
nici un parametru afişează prompterul standard
<promptstring> o combinaţie între $$, $t, $d, $p, $v, $n, $g, $l, $b, $q, $h, $e, $_

RECOVER </H> numefişier | u:
numefişier fişierul de refăcut
u: discheta / harddisk-ul de refăcut

RENAME | REN numefişier1 numefişier2
numefişier1 numele vechi al fişierului sau al directorului
numefişier2 numele nou

RESTORE </H> u_sursă: u:destinaţie: <numefişier> </S> </P>
u_sursă: unitatea de pe care se face restaurarea
u:destinaţie: unitatea pe care se restaurează
<numefişier> fişierul (fişierele) de restaurat
</S> se ţine seama şi de subdirectoare
</P> se cere acordul utilizatorului

RMDIR|RD <u:>cale
<u:>cale eliminarea unei căi

SET <var=<şir>>
nici un parametru afişează toate variabilele
<var=şir> atribuie variabilei var valoarea "şir"

SETUP </H>

SHARE </H> </X>
</X> dezactivează SHARE

SID </H>

SORT </H> </R> </+n>
</R> sortare în ordine inversă
</+n> sortează începînd cu coloana n

SUBST <u1: <u2:cale | /D>>
nici un parametru afişează lista de substituiri
<u1:<u2:cale>> unitatea u1: este substituită cu u2:cale
<u1: /D> se renunţă la substituirea făcută

SYS </H> u:
u: transferă sistem pe discheta (harddisk-ul) u:

TIME <oo.mm.ss>
nici un parametru afişează ora curentă şi cere o eventuală modificare
<oo.mm.ss> setează direct ora curentă la ora, minutul şi secunda specificate

TOUCH </H> <@>numefişier<...> </T:oo.mm.ss> </D:data> </F:E | J | U> </P> </R> </S>
<@>numefişier<...> (listă de) unul sau mai multe

segment

EXIT
nici un parametru iese din interpretorul de comenzi în uz

FASTOPEN </H>

FDISK </H> </D >
</D> permite FDISK şi pentru partiţii non-DOS

FILELINK </H> comandă <@><numefişier1> <numefişier2> <COMx:speed> </A> </D:data> </H> </M> </P> </R> </S> </U> </X>
comandă poate fi: DIRECTORY, DUPLICATE, RECEIVE, SETUP, SLAVE, QUIT, TRANSMIT
<@><numefişier1> (lista de) fişiere care urmează a se transmite
<numefişier2> destinaţie
<COMx:speed> x=interfaţa (1,2,3,4), speed = viteza de transmisie în Baud (11, 15, 30, 60, 12, 24, 48, 96, 19, 38, 56, 115)
</A> se transmit numai fişierele cu bitul de arhivare setat
</D:data> se transferă numai fişierele care au suferit modificări după data specificată
</H> se transmit şi fişierele sistem şi cele "ascunse"
</M> se transferă numai fişierele care au bitul de arhivare setat şi care nu există pe calculatorul destinaţie, sau există dar nu au setat bitul de arhivare
</P> cere acordul utilizatorului
</R> pe calculatorul destinaţie, şi fişierele read-only sînt suprascrise
</S> se ţine seama şi de toate subdirectoarele
</U> se transmit numai acele fişiere care pe calculatorul destinaţie nu există sau sînt mai noi decît versiunea de acolo
</X> nu se permite suprascrierea de fişiere pe calculatorul destinaţie

FIND </H> </B> </C> </F> </N> </S> </U> </V> "şir_căutat" <@>numefişier<...>
</B> afişarea se face într-un mod mai compact
</C> numără rîndurile în care apare şirul căutat
</F> afişează numai numele fişierelor în care apare şirul căutat
</N> afişează şi numărul curent al rîndurilor în care apare şirul căutat
</S> se ţine seama şi de toate subdirectoarele
</U> se face diferenţă între litere mici şi litere mari
</V> invers: se afişează numai rîndurile care **nu** conţin şirul specificat
"şir_căutat" şirul de caractere în cauză
<@>numefişier<...> (lista de) nume de fişiere în care să se caute

FORMAT </H> u: </S > </V> </1> </4> </8> </B> </T:xx> </X:x> </F:xxxx>

fişiere specificate
</T:oo.mm.ss> fixează o valoare pentru oră
</D:data> fixează o valoare pentru dată
</F:E | J | U> defineşte modul de prezentare al orei (Europa, Japonia, U.S.A.)
</P> cere acordul utilizatorului
</R> se operează şi asupra fişierelor protejate la scriere
</S> se ţine seama şi de subdirectoare

TREE </H> <u:> </B> </F> </G> </P> | <numefişier>
<u:> specifică unitatea
</B> nu mai afişează şi dimensiunea fişierelor
</F> afişează toate fişierele
</G> prezentare grafică
</P> paginare la afişaj
<numefişier> se iau în seamă numai fişierele ce corespund acestei specificări

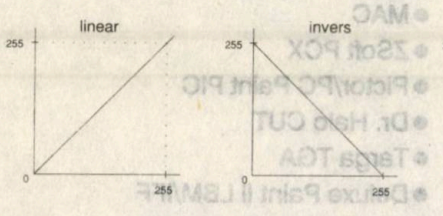
TYPE numefişier <;parola> </P>
numefişier<;parola> fişierul de afişat (eventual cu parolă)
</P> afişaj cu paginare

VERIFY <ON|OFF>
<ON|OFF> se va /nu se va face citire după fiecare scriere
nici un parametru afişează starea curentă a switch-ului

VOL <u:>
<u:> afişează numele discului (dischetei) de pe unitatea u:

XCOPY </Help> <@>numefişier <cale> </A> >/D:data> </E> </H> </M> </P> </R> </S> </V> </W>
</Help> afişează help-textul
<@>numefişier (listă de() fişiere de copiat
<cale>destinaţia
</A> se copiază numai fişierele cu bitul de arhivă setat
>/D:data> se copiază numai fişierele mai recente de data specificată
</E> se creează şi subdirectoare vide
</H> se copiază şi fişierele ascunse
</M> la fel ca /A, dar resetează bitul de arhivare după copiere
</P> cere acordul utilizatorului
</R> suprascrie şi fişiere protejate la scriere
</S> se ţine seama şi de subdirectoare
</V> cu verificare
</W> se aşteaptă schimbarea dischetei (pentru copieri folosind două dischete pe aceeaşi unitate)

DIP ce lucrează în timp real. Sub numele de LUT înţelegem o tabelă de transformare, care transformă valoarea intensităţii luminoase a unui pixel de intrare, într-o altă valoare. Transformarea se poate face la nivelul fiecărui punct din imagine. Octetul livrat de convertorul A/D, reprezintă valoarea de intrare în tabelă. Exemple tipice de LUT sînt: transformarea liniară, transformarea inversă, transformarea logaritmică.



Offset

Variaţia nivelului de bază a semnalului video analogic.

Pixel

Cel mai mic element de memorare a unei imagini numerice care poate fi controlat de sistemul de redare discretă. Un pixel se poate adresa pe baza coordonatelor sale orizontale şi verticale.

Pixelsynchron

De obicei, semnalul video generat de camera de luat vederi este digitizat de frame grabber cu un tact de 10 pînă la 15 MHz. În acest caz însă nu putem preciza cu exactitate corespondenţa dintre punctele de imagine a camerei CCD şi punctele imaginii digitizate. Interfeţele FG cu posibilitate de explorare multiplă - VARIABLE SCAN, pot utiliza ca tact de digitizare, tactul de pixel (PIXELCLOCK) livrat de camera CCD. Astfel punctele imaginii digitizate devin "PIXELSINCRONE" cu punctele imaginii generate de cameră.

Pixel pătrat - Square Pixel

O cameră de luat vederi de uz comercial, conţine un cip CCD, care are un raport de 4:3 între lăţimea şi înălţimea imaginii. Dacă de exemplu, un sistem de prelucrare de imaginii preia cu această cameră imaginea unui CERC, în memoria de imagine vom găsi digitizată o ELIPSĂ, cu care vom putea lucra mai departe doar dacă facem corecţiile de rigoare. Dacă, însă, vom utiliza un frame grabber ce foloseşte pixeli pătraţi, atunci semnalul generat de cameră va fi digitizat sincron cu un tact de 15 MHz (în loc de 10 MHz). Rezoluţia mai mare pe orizontală permite ca raportul dintre lăţimea şi înălţimea imaginii obţinute să fie în concordanţă cu raportul imaginii oferite de cameră, scutindu-ne astfel de prelucrări ulterioare.

Timp real - Real Time

Despre o operaţie care se poate face într-un timp mai mic decît ciclul unui cadru, se spune că a fost făcută în timp real. În formatul video european CCIR, un ciclu de un cadru este de 1/25 secunde = 40 milisecunde (frecvenţa cadre).

Saturaţie - Saturation

Într-un sistem COLOR, saturaţia reprezintă "puritatea" unei culori sau cît de slab sau tare apare o culoare în imagine.

Spin

Fenomen, ce există în domeniul prelucrărilor de imagini, caracterizat prin întîrzierea unui pixel în timp, ce are ca efect neînscrierea lui la aceeaşi adresă în memoria de imagine. Spinul se poate produce atît pe orizontală cît şi pe verticală.

Semnal de sincronizare

Conţine informaţia de TIMING, cu ajutorul căreia se poate comanda un monitor TV sau o cameră video. Un semnal de sincronizare are două componente:

- -semnalul de sincronizare linii şi
- -semnalul de sincronizare cadre.

Între aceste semnale se livrează monitorului, 625 de linii şi 2 semicadre, din care însă numai 576 de linii sînt utilizabile din imaginea video în sine pentru că restul liniilor se folosesc ca linii de întoarcere a spotului electronic la început de linii şi la început de cadre.

Explorare multiplă - Variable Scan

Operaţie de citire a unor semnale video, care nu sînt standard cu nici o normă video existentă. Ca semnal de intrare, se utilizează un semnal analogic sau digital, un tact de pixel, un semnal de alternare a liniilor şi un semnal de deblocare cadre.

Cîştig - Gain

Amplificarea unui semnal analogic cu scopul de al digitiza. Se pot utiliza diferite amplificări.

Rezoluţie pe verticală - Vertical Resolution

Numărul vizibil de linii pe verticală al unei imagini redate dintr-un rastru al unui sistem grafic cu explorare electronică, sau numărul axelor verticale ale unui display. În sistemele video, ea reprezintă numărul liniilor de test orizontale ce se pot genera de la o cameră sau reproduce pe un monitor (tipic 70% din numărul total de linii ale rastrului).

"Tremurat" - Jitter

Instabilitatea unui semnal în amplitudine şi/sau fază.

(ing. Kalló Tibor)

micro ATCI
Reprezentantul oficial în România al firmei Leutron Vision AG din Zürich, Elveţia, specializată în prelucrarea digitală a imaginilor, vă stă la dispoziţie !
Micro ATCI C.P. 64, O.P. 1, RO-4300 Tîrgu Mureş, Tel. 954/20521, Fax 954/35208

Formate grafice

De această dată vă vom prezenta modul în care puteţi transporta grafice între diferite programe şi sisteme de calcul.

În ultimul număr al revistei **"if"** v-am prezentat diferitele formate grafice folosite de diferitele programe de grafică. În partea a doua ne vom ocupa de conversiile formatelor grafice. Pentru aceasta, de regulă, sînt necesare programe utilitare, care să rezolve aceste conversii. Pe piaţă sînt ofertate pentru acest scop următoarele programe:

- Hijaak
- XPort
- Bridge
- VPIC
- GCP

Hijaak

"Hijaak" este un produs al firmei North American Software şi costă cca. 420 DM. Pe lîngă modulul de "fotografiere" al ecranului şi de prelucrare a imaginii obţinute, el este şi un program puternic şi confortabil de conversie de formate grafice "bitmap". Posibiltăţile de conversie sînt prezentate în caseta alăturată.

XPort

Un alt domeniu acoperă produsul "XPort" al firmei Micrografx. El costă 500 DM şi lucrează sub suprafaţa utilizator Windows 3.0. XPort poate fi utilizat pentru conversia formatelor grafice vectoriale. Sînt posibile cinci formate de intrare, ca şi de ieşire, aceasta însemnînd că sînt posibile conversii între toate cele cinci formate în ambele direcţii. Aceste formate sînt:

- DRW (Designer)
- GEM (GEM Draw, GEM Artline)
- CGM
- Mac Pict (versiunile I şi II)
- DXF (programe CAD)

O caracteristică a lui XPort este prezenţa pe ecran în timpul conversiei a unei rigle pe care este arătat în procente cît a fost convertit deja. Dacă în timpul conversiei sînt întîlnite în formatul de intrare elemente de desen necunoscute formatului de ieşire (de ex. un degrade de culori dintr-un desen DRW obţinut cu Designer, care trebuie convertit în format DXF pentru Autocad) XPort alege în mod automat acele elemente de desen ale formatului de ieşire care sînt cele mai apropiate de cele din formatul de intrare.

Bridge

Programul "Bridge" este un produs al casei germane de soft GCA din Ludwigsburg şi face parte din pachetul "Windows-Tools", o colecţie de utilitare pentru Windows. Bridge este conceput în mod special pentru transferul de date între GEM şi Windows şi oferă următoarele posibilităţi de conversie:

- GEM în PIC (Micrografx In-A-Vision poate fi citit de Designer)
- IMG în MSP
- IMG în PCX
- DOC în WRI (din GEM Write în Windows Write).

VPIC 2.8

De cu totul altă natură este programul "VPIC" un produs Public-Domain de Bob Montgomery care poate fi copiat şi dat mai departe fără restricţii. El poate fi folosit la vizualizarea diferitelor formate bitmap pe ecran şi la prezentarea unor secvenţe de imagini (diafilme) care să ruleze automat. El presupune existenţa cel puţin a unei plăci grafice EGA sau VGA şi se descurcă cu o sumedenie de plăci grafice Super VGA. Programul VPIC poate vizualiza următoarele formate grafice:

- GIF
- MAC
- ZSoft PCX
- Pictor/PC Paint PIC
- Dr. Halo CUT
- Targa TGA
- Deluxe Paint II LBM/IFF
- Colorix/EGA Paint SCx

Conversia într-un alt format decurge foarte simplu. În timp ce imaginea este afişată pe ecran se apasă o anumită combinaţie de taste şi imaginea este stocată automat în formatul de ieşire dorit. Sînt posibile următoarele formate de ieşire:

Format de ieşire	Format destinaţie (Hijaak)
DPaint LBM/IFF	DPaint LBM/IFF
Compuserve GIF	Compuserve GIF
Fax Complete PC	Fax Complete PC
Fax Gamma	Fax Gamma
Fax Generic	Fax Generic
Fax Intel (B/W PCX)	Fax Intel (B/W PCX)
Fax Jetfax	Fax Jetfax
Fax JT	Fax JT
Fax OAZ	Fax OAZ
Fax Ricoh	Fax Ricoh
Fax Teli	Fax Teli
Fax Worldport	Fax Worldport
GEM IMG	GEM IMG
Dr. Halo CUT	Dr. Halo CUT
HP Laserjet PC	LHP Laserjet PCL
HP Plotter PGL	HP Plotter PGL
Inset PIX	Inset PIX
Lotus PIC	Lotus PIC
Macintosh Macpaint	Macintosh Macpaint
Macpaint (No header)	Macpaint (No header)
Microsoft Paint (Ver 1)	Microsoft Paint (Ver 1)
Microsoft Paint (Ver 2)	Microsoft Paint (Ver 2)
Newsmaster	Newsmaster
PC Paintbrush	PC Paintbrush
Printmaster	Postscript
Text Printmaster	
TIFF (Compressed)	Text
TIFF (Uncompressed)	TIFF (Compressed)
Wordperfect WPG	TIFF (Uncompressed)

- PCX
- PIC (PC-Paint respectiv Pictor, programul de pictură livrat împreună cu programul de animaţie Grasp)
- GIF
- Colorix SCx
- TGA (Targa)
- CUT (Dr. Halo)

GCP

Tot pentru conversia de grafice bitmap este gîndit şi programul GCP, un program shareware al firmei Synergistic Enterprises din America, care lucrează sub Windows 3.0. Taxa de înregistrare, în cazul în care vă decideţi să utilizaţi programul după ce l-aţi testat, este de 40$. Programul recunoaşte atît la intrare cît şi la ieşire următoarele formate:

- MSP
- Macpaint
- GIF
- HP PCL
- Windows Bitmap (BMP)
- Windows Cursor
- Windows Icon

Ultimele două formate ar putea fi interesante mai ales pentru proiectanţii care dezvoltă aplicaţii sub Windows.

Următoarele programe pe care vi le prezentăm, şi anume "Corel Draw", "Micrografx Designer" şi "Art & Letters" propriu-zis nu sînt programe de conversie ci pachete complexe de desenare şi design de cel mai înalt nivel. Toate oferă însă, pe lîngă funcţiile proprii, posibilitatea de a exporta şi importa un întreg şir de formate grafice. Programele enumerate lucrează orientat obiect; ceea ce înseamnă că ele nu privesc imaginile grafice ca pe un ansamblu de puncte pe monitor, ci ca pe nişte colecţii de obiecte, cum ar fi: cercuri, linii, dreptunghiuri. Ele pot însă importa şi exporta şi formate bitmap.

Pagemaker	Ventura 2.0 cu extensia profesională
WRI, Windows Write	ASCII
TXT,Word	8-Bit-ASCII
Wordperfect	Wordstar 3
Euroscript II	Multimate
Wordstar 3.4	Word
ASCII	Writer
Smart ASCII	WS 4.0/5.0
DCA	Xywrite
DRW, Windows Draw	DCA
PIC, In-A-Vision	Wordperfect 5.0
WMF	PRN to Table
DRW Micrografx Draw	GEM
PCX	WMF
TIFF	Auto-CAD SLD
Datacopy	Lotus PIC
Mouse Systems PC Paint	Video Show
Macpaint	Mac Pict
Auto-CAD	ADI CGM
EPS	Postscript
HPGL	HPGL
Lotus PIC	
Lotus WK1	IMG
dBase DBF	PCX
Macpaint	TIFF

Micrografx Designer 3.01

Import

- DRW (pentru cuplarea a două desene)
- GRF (Micrografx Charisma)
- PCX (şi color)
- TIFF (inclusiv import de imagini color în format 24 de biţi - maxim 16.7 milioane culori)
- TXT (text ASCII normal)
- WMF
- DXF (Autocad, etc.)
- CGM
- GEM (Draw şi Artline)
- PCT (Macintosh Pict I & II)

Export

- DRW (versiunile 1.0 pînă la 3.0)
- EPS (la alegere cu sau fără antet TIFF)
- HPGL
- PCX
- PIC (In-A-Vision)
- TIFF (pînă la color 24 de biţi)
- WMF
- DXF
- CGM
- GEM
- PCT

Corel Draw 1.2

În versiunea 1.2 care lucrează sub Windows şi Corel Draw oferă o multitudine de formate de export şi import. Şi el poate trasa automat imagini bitmap.

Import

- CDR (format original Corel Draw)
- TIFF
- PCX
- DXF
- EPS (numai format Adobe Illustrator şi formatul EPS al modulului de trasare Corel Trace)
- GEM
- CGM
- HPGL
- IBM-PIF
- Lotus PIC
- Macintosh Pict

Export

- EPS (la alegere cu/fără antet TIFF)
- WMF
- PCX
- TIFF
- DXF
- GEM
- CGM
- HPGL
- IBM-PIF
- Adobe-Illustrator (.AI)
- Macintosh Pict
- SCODL (pentru exponatoare matriciale)
- Video-Show PIC
- WPG (Wordperfect Graphic)

Pentru posibilităţile de import/export vezi şi tabelul comparativ între Designer 3.01 şi Corel Draw 2.0 din articolul următor.

Art & Letters 3.0

Import

- EPS
- TIFF (şi color 24 biţi TIFF)

- WMF
- PIC (Lotus)
- ASCII (text)
- DIA (format propriu al producătorului)
- TGA (Targa)
- GIF
- PCX

Export

- EPS (Standard sau Adobe Illustrator cu/fără antet TIFF)
- CGM
- TIFF (max. 256 culori)
- SCODL
- WMF
- PC-MT (format ecran propriu).

Dacă dispuneţi de unul din programele enumerate atunci aveţi la îndemînă o "maşină de conversie" aproape universală. Limitări există desigur. Programe cum ar fi Designer, Artline, Arts & Letters sau Corel Draw tratează obiectele care nu sînt create din linii drepte (poligonale) ci din linii curbe prin intermediul aşa-numitelor curbe Bezier. Dacă o astfel de imagine este listată pe o imprimantă Postscript, sau este exportată în format EPS, atunci aceste linii curbe vor fi reproduse conform originalului. Dacă aceste imagini sînt exportate într-un format care nu recunoaşte curbele Bezier, acestea vor fi înlocuite cu curbe poligonale. La conversia în sens invers se va întîmpla acelaşi lucru. Din această cauză în timpul conversiei pot fi pierdute detalii.

Windows Clipboard

O altă modalitate de conversie elegantă este utilizarea "magaziei intermediare" (clipboard) a lui Windows. În ea pot fi memorate atît texte cît şi imagini bitmap sau vectoriale. Orice program care lucrează sub Windows are şi un meniu "Prelucrare" care oferă opţiunile "Secţionare", "Copiere" şi "Inserare". Astfel cu mouse-ul poate fi marcat un obiect din programul grafic şi să se aleagă opţiunea "Copiere", obiectul va fi copiat în clipboard de unde va putea fi preluat şi inserat într-o imagine aparţinînd altui program.

Răsplata ostenelii

După ce aţi realizat o imagine grafică, cu sau fără conversie, urmează încă un pas: listarea pe o imprimantă sau pe un exponator, sau prelucrarea cu un alt program. De regulă urmează o prelucrare cu un program DTP (Desktop Publishing), programe care ştiu citi şi importa o serie întreagă de formate grafice. În caseta alăturată aruncăm o privire asupra celor mai importante programe DTP din sectorul PC şi Macintosh: Pagemaker şi Ventura Publisher.

(R.M.)

Grafică: de la un sistem la altul

Dacă se doreşte transferul unor fişiere cu grafică între diferite sisteme de calcul, problemele care se ivesc, de obicei, sînt mari deoarece, de ex., dischetele nu se pot schimba prea uşor între Amiga şi Mac, deoarece cele două calculatoare folosesc formate diferite. Soluţii există totuşi: atît PC, Amiga, ST cît şi Macintosh posedă o interfaţă serială standardizată, prin care pot fi cuplate, de ex., la un modem. Se poate renunţa însă la legătura via telefon şi să se cupleze direct cele două calculatoare prin intermediul unui cablu. Nu mai este necesar apoi decît un program de transfer, adesea denumit şi program terminal.

PC - Macintosh

Pentru schimbul de date între PC şi Macintosh există următoarele posibilităţi:

» a) Pentru modelele Macintosh există un lector de dischete de 5,25", produs de Apple, care poate citi şi scrie dischete MS-DOS de 360 KBytes. Limitare : dimensiunea maximă a unui fişier 360 KBytes.

» b) modelele Macintosh mai noi, începînd de la Mac IIi dispun de unităţi de dischete de 3,5" care pot citi şi scrie şi dischete MS-DOS de 1,44 MBytes.

» c) pentru fişiere foarte mari şi un transfer rapid poate fi utilizat programul Maclink al firmei Travelling Software.

PC - Amiga

Pe lîngă transferul prin cablu pe interfaţa serială mai pot fi utilizate şi următoarele posibilităţi de transfer: dacă se dispune de un Amiga 2000 cu placă de legătură AT sau PC, atunci pot fi folosite programe speciale de transfer. Pe lîngă acestea mai există şi mici programe utilitare, aşa-numite DOS-Devices, care permit lectorului de dischete Amiga să citească dischete în format MS-DOS.

PC - ST

Atari ST poate citi dischete de 3,5" cu o capacitate de 720 KBytes şi dischete de 5,25" cu capacitatea de 360 KBytes, numai dacă dischetele au fost formatate pe PC. Altfel sînt necesare şi în acest caz programe speciale pentru ST.

Alte legături

Deoarece formatele dischetelor sistemelor Mac, Amiga şi ST sînt foarte diferite, pentru transferul de fişiere între ele nu poate fi folosită decît legătura prin interfaţa serială.

Designer 3.01

Numele ne arată deja că "Designer 3.01" vrea să fie mai mult decît un program de desen pentru grafică vectorială.

Furnitura de livrare a programului Designer 3.01 conţine: 5 dischete de 5,25", un manual de introducere, unul de utilizare şi o casetă video pentru studiu.

Instalarea cere un timp mai îndelungat. Deoarece procedura de instalare trebuie să pornească de la suprafaţa DOS trebuie să se renunţe la ajutorul cu care ne-a obişnuit Windows. Doar cu manualul de instalare se poate obţine claritate asupra diverselor operaţiuni de instalare. Cine are nevoie de biblioteca Clip-Art trebuie să se aştepte la un timp de instalare şi mai îndelungat, deoarece toate fişierele au fost comprimate cu utilitarul LHARC, un program de arhivare care atinge rate de comprimare înalte, dar care nu mai este "state of the art".

După ce s-a terminat munca de instalare se poate începe lucrul cu Designer. La pornire în partea stîngă a ecranului se afişează o listă de simboluri grafice (iconuri) din care se pot alege cele mai importante instrumente de desen. Toate celelalte funcţiuni sînt cuprinse în meniuri pull-down. O facilitate deosebită o constitue posibilitatea preluării unor puncte de meniu şi adăugarea lor la lista de iconuri. Cel care foloseşte o funcţie foarte des o poate instala şi pe tasta din dreapta a mouse-ului.

Deservirea programului nu este chiar atît de simplă cum pare la prima vedere. Deoarece meniurile sînt foarte voluminoase, diversitatea de funcţii zăpăceşte la început. Cu toate că funcţia de help on-line este integrată, Micrografx a renunţat din păcate la lecţiile introductive (Tutorial). În schimb se livrează un manual de introducere în care se explică primii paşi cu Designer cu cîteva exemple de grafică.

În timpul testelor cu Designer, utilizatorul îşi va da repede seama că proiectantul nu s-a putut hotărî între deservirea cu mouse sau cu tastatura. Prelucrarea obiectelor este posibilă şi folosind doar mouse-ul, dar se încetineşte procesul de lucru. Doar combinînd tastele SHIFT şi CTRL cu mouse-ul se poate lucra eficient. Pentru aceasta ar trebui învăţate pe dinafară toate combinaţiile tastatură-mouse, ceea ce la o suprafaţă SAA (Systems Applications Architecture) ar trebui evitat. Pe de altă parte lucrul cu Designer dă satisfacţii destul de mari, deoarece tehnica WYSIWYG (ceea ce este afişat pe ecran să se obţină şi la imprimantă - What you see is what you get) este pe deplin sprijinită. Atunci cînd pe ecran se îngrămădesc prea multe obiecte, faptul că imaginea este împrospătată după cea mai mică modificare poate deveni deranjant.

Un criteriu important în alegerea unui soft de grafică este volumul şi calitatea funcţiilor de prezentare. Fireşte că în Designer sînt integrate toate funcţiile standard (trasare linii, dreptunghiuri, cercuri, etc., rotirea obiectelor, întinderea lor, etc.). În plus sînt oferite şi posibilităţi de manipulare a curbelor Bezier. Aceste curbe permit realizarea oricărui tip de legătură între două puncte: fie prin linii simple, parabole, semicercuri, sau curbe aleatoare. Designer permite transformarea obiectelor în curbe Bezier sau dezvoltarea lor din astfel de curbe. Începînd de la manipularea punctelor de ancorare şi de control şi pînă la "răsturnarea" curbelor, totul a fost gîndit.

Un alt punct forte este introducerea şi manipularea textelor. Designer poate folosi fie fonturile proprii, liber scalabile, fie fonturile Windows, fie fonturile proprii imprimantei cu care se lucrează, ceea ce în cazul utilizării unei imprimante Postscript poate deveni interesant.

Fonturile proprii Designer sînt însă relativ puţine în cadrul pachetului livrat. Este păcat deoarece doar aceste fonturi pot fi transformate apoi în curbe Bezier şi manipulate.

Un alt punct vulnerabil al lui Designer este volumul funcţiilor de prelucrare. Există totuşi toate funcţiunile standard şi în cadrul meniului "grupare" este inclus un instrument puternic pentru manipularea unei întregi grupe de obiecte. Designer posedă şi operaţiile de bază pentru măşti şi colorare cît şi o funcţie pentru manipularea liniilor. Cu aceasta din urmă s-ar putea obţine chiar şi efecte de caligrafie, dacă ar funcţiona corect... Din păcate acest lucru nu se întîmplă întotdeauna.

Funcţii (pseudo) 3D sau "Envelopes" nu există, totul trebuie rezolvat manual. Astfel se pot realiza foarte greu efecte de spaţiere.

Împreună cu Designer se livrează două pachete auxiliare care uşurează mult munca: în primul rînd biblioteca Clip-Art care este foarte cuprinzătoare. Utilizatorului îi sînt oferite chiar şi funcţii de administrare a ei. În al doilea rînd amintim funcţia Trace cu care se pot vectoriza imagini bitmap. Designer recunoaşte un număr foarte mare de formate, chiar şi imaginile color putînd fi transformate în grafică vectorială.

În pachetul de livrare mai sînt conţinute şi "MGX Slide" pentru sesiuni de prezentare (slideshow), "MGX Print" un utilitar de tipărire şi "Telegrafx" cu care se pot transfera imaginile via modem spre exponatoare.

Corel Draw 2.0

După apariţia lui Corel Draw 1.2 se punea întrebarea: "Ce s-ar mai putea îmbunătăţi la acest produs ?". Şi pentru că nu se cunoştea răspunsul produsul a fost acoperit cu laude. Producătorii de la Corel Systems nu s-au mulţumit însă cu atît şi au produs versiunea 2.0, care este tocmai răspunsul la întrebarea anterior pusă: "Aproape totul poate fi îmbunătăţit !". Şi tocmai acest lucru îl demonstrează versiunea 2.0.

Această versiune se potriveşte în întregime resurselor lui Windows 3.0. Imaginile color sînt afişate (cu o placă VGA desigur) în 256 de culori reale, o adevărată bucurie pentru ochi. Sumedenia noutăţilor versiunii 2.0 depăşesc limitele acestui test.

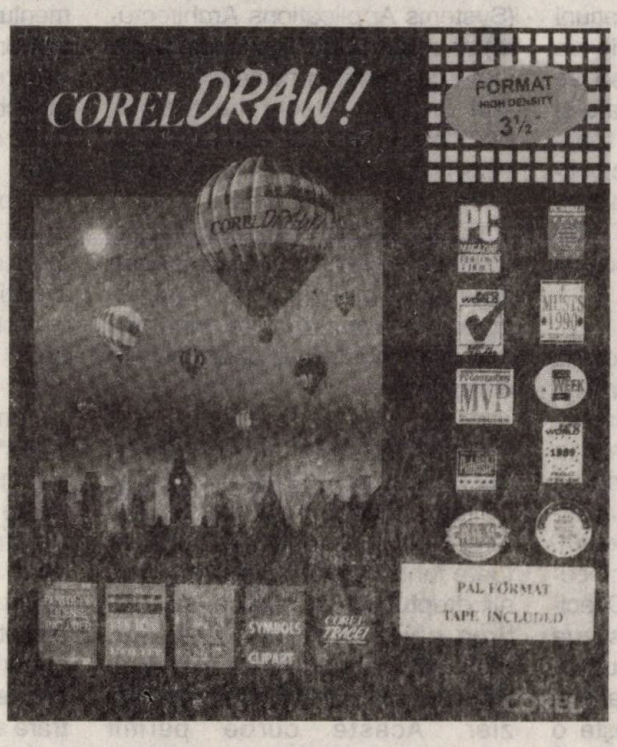
Vom da totuşi cîteva exemple: de-a dreptul revoluţionar este noul meniu "Efecte". În acest meniu există funcţiuni care permit pentru prima dată lucrul în 3 dimensiuni. Acest lucru nu se întîmplă ca şi la programele CAD (Computer Aided Design) prin introducerea coordonatelor spaţiale, uluitorul efect de spaţialitate este realizat cu ajutorul aşa-ziselor "puncte de fugă".

Tot la fel de surprinzătoare sînt şi efectele care pot fi obţinute cu ajutorul funcţiei "Envelope": cu ajutorul lui obiectele pot fi "strivite" în diferite forme, astfel încît, de ex., să se imprime un text pe suprafaţa unei perne de pe o sofa, sau să se "îndoaie" un anumit nivel din spaţiu. Aceste forme, în parte confuze, pot fi trase apoi deasupra altor obiecte. Acest lucru este valabil şi pentru constelaţia punctelor de fugă ce pot fi create cu ajutorul funcţiei "Perspectivă".

La fel de noi sînt şi cîteva posibilităţi de prelucrare a textelor, care n-au mai fost văzute pînă acum în nici un program grafic. Astfel, de ex., dintr-un grafic pot fi extrase anumite pasaje de text, pot fi modificate cu un procesor de texte adecvat şi apoi pot fi inserate la loc în grafic.

Modificări mari au fost făcute şi în modul de tratare al culorilor. Pe lîngă introducerea a două modele color noi RGB şi HSB, acum se poate lucra şi cu o paletă reală de 256 de culori. Rezultatele obţinute sînt surprinzătoare, imaginile rezultate fiind cvasi-reale. Supărătoare este doar viteza cu care se obţine o imagine, chiar şi pe un 486. Dar la acest capitol nu stau mai bine nici alte aplicaţii grafice Windows.

Îmbucurător de ridicată este viteza cu care sînt încărcate fişierele de text în boxele de dialog. Încărcarea unui text dintr-un fişier selectat într-o fereastră de vizualizare se face aproape în timp real, dar funcţionează doar cu fişiere care au fost create din Corel Draw 2.0. Încă un pas înainte s-a făcut cu noul program de arhivare "Mosaic". Este vorba de un program de arhivare de sine stătător pentru arhivarea de imagini.



În biblioteca Clip-Art sînt furnizate peste 750 de grafice, grupate în categorii şi arhivate cu Mosaic. Se cîştigă astfel spaţiu pe disc dar se pierde mult din viteza de lucru, chiar şi pe un 486 durînd destul de mult pînă cînd este extras conţinutul arhivei pe ecran. Tot la fel de mult durează şi pînă cînd este extras un grafic individual din bibliotecă. Şi Corel Draw utilizează pentru acest scop programul LHARC, şi nu într-o versiune Windows ci în vechea versiune DOS.

De remarcat sînt şi programele auxiliare "Corel Trace" şi "Win Boss", cunoscute încă din versiunea anterioară dar înnoite şi ele. Corel Trace poate vectoriza acum şi imagini bitmap color, care pot fi prelucrate fie în 256 de culori fie se poate face o anumită reducere de culori. Utilitarul Wfn-Font poate converti acum fonturile Corel în format Adobe Type 1. Se obţin astfel dintr-o lovitură 153 de seturi de caractere vectorizate de primă clasă, care pot fi utilizate şi pentru alte aplicaţii care utilizează fonturi Adobe. Dacă luăm în considerare şi noile facilităţi de a memora grafice în format Wfn rezultă că sistemul poate fi utilizat ca editor de fonturi, printre altele, pentru Adobe Type Manager.

În rest ar fi nenumărate alte îmbunătăţiri şi înnoiri de amintit, cum ar fi: paleta de culori şi biblioteca de simboluri editabile, linii de ajutor liber definibile, funcţie de salvare/restaurare (backup), posibilitatea de a instala o funcţie pe tasta din drepta mouse-ului, sau noile opţiuni pentru efecte de degrade.

Corel Draw 2.0 este un produs foarte puternic, care nu va avea prea curînd o concurenţă serioasă.

(I.M.)

Nume	Designer 3.01	Core Draw 2.0
Preţ aprox. (DM)	2540	1700
Instalare / Introducere		
Procedura	DOS	DOS
Lecţii introductive	manual	manual, video
Funcţii de ajutor	da	da
Manuale	2	7
Introducere pe casetă video VHS	da	da
Prietenos cu utilizatorul		
Suprafaţă SAA	da	da
Instalare funcţie pe tasta dreaptă a mouse-ului	da	da
Preluare unui punct de meniu ca simbol în lista afişabilă	da	da
Obiecte de bază		
Linie	da	da
Dreptunghi	da	da
Dreptunghi rotunjit	da	da
Arc de cerc	da	da
Diagramă cerc	da	nu
Elipsă	da	da
Arc de elipsă	da	da
Transformări		
Rotiri	da	da
Înclinări	da	da
Scalări	da	da
Oglindiri	da	da
Efecte		
Envelopes (transformare neliniară)	nu	da
Perspectivă (efect 3D)	nu	da
Extrude (corpuri 3D pentru anumite obiecte)	nu	da
Amestec de culori (evoluţia obiectului cu potrivirea culorilor)	nu	da
Amestec de forme (evoluţia obiectului cu potrivirea formei)	nu	da
Ordonarea obiectelor		
Prim plan	da	da
Fundal	da	da
Un nivel înainte	nu	da
Un nivel înapoi	nu	da
Schimbarea succesiunii	nu	da
Grupare/ eliminare grupare	da	da
Combinare de obiecte	da	da
Secţionare obiecte	da	da
Aliniere (ordonare geometrică)	da	da
Afişare pe ecran		
Zoom (lupă)	da	da
Toate obiectele	nu	da
Schiţă fără culori	da	da
Previzualizare completă ecran	nu	da
Rastru de ajutor	da	da
Linii de ajutor	nu	da
Cursor cruce	da	da
Liniar	da	da
Indicator de stare	nu	da
Paleta de culori	nu	da
Opţiuni suprafaţă		
Umplere cu obiecte vectoriale	da	da
Umplere cu bitmap	da	da
Degrade liniar	da	da
Degrade radial	da	da
Offset pentru sursa de lumină la degrade	da	da
Opţiuni linie		
Linie întreruptă	da	da
Linie invizibilă	da	da
Terminatori de linie selectabili	da	da
Caligrafie	da	da
Punct final selectabil	da	da
Manipulări Bézier		
Ştergere puncte de sprijin	da	da
Inserare puncte de sprijin	da	da
Separare linii/curbe	da	da
Cuplare linii/curbe	da	da
Conversie curbe în linii	da	da
Conversie linii în curbe	da	da
Vectori neliniari	da	da
Vectori liniari	da	da
Vectori simetrici	nu	da
Aliniere puncte de sprijin	nu	da
Tratare culori		
Nr. max. culori curate	16	256
Paletă Pantone	nu	da
Paletă CMYB	da	da
Paletă HSB	da	da
Paletă RGB	da	da
Funcţii text		
Nr. fonturi	13	153
Import text	da	da
Export text	nu	da
Paragraf	da	da
Aliniere	da	da
Spaţiere linii	da	da
Spaţiere caractere	da	da
Import grafic		
AI	nu	da
BMP	nu	da
CGM	da	da
DRW	da	nu
DXF	da	da
EPS	nu	da
GEM	da	da
GRF	da	nu
HPGL	nu	da
PCT	da	nu
PCX	da	da
PIC	da	da
PICT	nu	da
PIF	nu	da
SCODL	nu	da
TIFF	da	da
Video Show	nu	da
WFM	nu	da
WMF	da	da
WPG	nu	da
Export grafic		
AI	nu	da
CGM	da	da
DRW	da	nu
DXF	da	da
EPS	da	da
GEM	da	da
HPGL	da	da
PCT	da	nu
PCX	da	da
PIC	da	nu
PICT	nu	da
PIF	nu	da
SCODL	nu	da
TIFF	da	da
Video Show	nu	da
WFM	nu	da
WMF	da	da
WPG	nu	da
Funcţii de vectorizare		
Nr. max. culori	16	256
Reducţie de culori	nu	da
Autotrasor	nu	da
Trasare Inline	nu	da
Trasare Outline	da	da
Parametrii de trasare programabili	limitat	da
Import de fonturi		
Bitstream	da	da
URW	da	nu
Digifonts	nu	da
Readable Postscript	nu	da
ZSoft Type Foundry	nu	da
Substitution	nu	da
Agfa/Compugraphic Type Director	nu	da
Adobe Type 1	nu	da
Export de fonturi		
ZSoft Type Foundry	nu	da
Adobe Type 1	nu	da
Programe auxiliare		
Manager de vizualizare fişiere	nu	da
Print-Manager	da	nu
Program de vectorizare	nu	da
Program de conversie de fonturi	nu	da
Program de transmisie la distanţă	da	nu

Transparenţă şi polimorfism

OOP - Partea a doua

În a doua parte a cursului nostru ne vom ocupa de polimorfism, care alături de moştenire, este o caracteristică extrem de importantă a obiectelor.

Modalităţile de expresie ale limbii umane sînt foarte variate. Examinaţi următoarele propoziţii scurte.

"Producţia stă."
"Ceasul stă."
"Locatarul stă."

Cu ajutorul cuvîntului "stă", în funcţie de context, se pot obţine mai multe semnificaţii. Este utilizat deci unul şi acelaşi cuvînt pentru a descrie situaţii diferite. Semnificaţia cuvîntului "stă" devine clară abia în legătură cu situaţia pe care o descrie. Desigur vă întrebaţi, ce are a face acest lucru cu programarea orientată obiect ? Are şi încă foarte mult, deoarece şi în cazul OOP aveţi posibilitatea de a obţine reacţii diferite pentru o aceaşi sarcină. Această caracteristică foarte importantă a OOP este denumită polimorfism.

După cum ştim obiectele îşi pot lăsa moştenire însuşirile altor obiecte noi. Metodele preluate de la obiectul mamă pot fi însă, la nevoie, înlocuite cu metode noi avînd acelaşi nume şi aceeaşi listă de parametri. Tocmai această caracteristică OOP este desemnată ca polimorfism. La un apel prin intermediul obiectului fiu în locul metodei vechi va fi folosită noua metodă. Acest lucru nu este nou. Şi pînă acum, în Pascal, rutinele predefinite puteau fi înlocuite cu alte rutine avînd acelaşi nume. Nou este însă faptul că metodele înlocuite nu se pierd şi pot fi şi ele accesate în continuare.

De aici rezultă cîteva posibilităţi practice. Mai întîi puteţi să produceţi pentru aplicaţia d-voastră obiectele avînd metodele generale. Din acestea puteţi defini apoi obiectele speciale, care să folosească metodele special definite pentru ele.

În al doilea rînd, un obiect fiu va putea folosi, în funcţie de necesităţi, metodele moştenite de la obiectul mamă, sau metodele nou definite. În cazul în care programaţi folosind doar metode statice, în momentul compilării trebuie să fie clar ce cod va fi înglobat, aşa cum vă prezentăm şi în fişierul STR_ROUT.PAS. Folosim, în acest caz, cunoştinţele pe care le avem despre obiecte şi facem definiţiile într-un Unit. Declarăm obiectul nostru în partea Interface, iar tot restul în partea Implementation. În acest mod întărim încapsularea. Tipul de obiecte "BS" defineşte datele şi metodele unei prelucrări simple a unui şir de caractere. Pot fi definite, desigur, mai multe metode, dar cele şase definite sînt suficiente pentru demonstrarea polimorfiei. Pe noi ne interesează metoda "scrie_cuvint". Ea scrie şirul de caractere introdus direct în memoria de împrospătare a ecranului. Pentru text şi fundal vor fi folosite valorile pe care variabila predefinită "TextAttr" le-a stabilit în Unit-ul "crt". La un moment dat s-ar putea ca metodele astfel definite să nu vă mai fie suficiente. De exemplu atunci cînd doriţi scrierea unui şir de caractere de o anumită culoare.

Definiţi deci un nou obiect. În fişierul LIST_8.PAS l-am denumit "BS2". Tot acolo am definit două metode pentru stabilirea culorii textului şi fundalului. Interesantă este realizarea metodei tocmai moştenită de la obiectul mamă "scrie_cuvint". Ea stabileşte culorile înainte de scriere şi apoi restaurează vechile valori. Ceea ce se face între aceste două operaţii nu este nou. Codul a fost doar preluat.

Gata ! În funcţie de tipul obiectului, puteţi accesa acum metoda veche sau pe cea nouă. Examinaţi codul cu atenţie şi veţi observa că metoda propriu-zisă de scriere se aseamănă foarte mult cu metoda veche. Codul metodei "BS2.scrie_cuvint" poate fi deci înlocuit cu codul descris în fişierul STR_ROUT.PAS. Încercaţi ! Metoda veche poate fi inserată în noua metodă. Pe de o parte acest lucru este comod şi economiseşte timp. Pe de altă parte ne asigurăm în acest fel că toate iniţializările obiectului mamă vor fi preluate.

Şi mai există încă un avantaj: dacă la un moment dat modificaţi obiectul mamă, nu trebuie să modificaţi şi obiectele definite cu ajutorul lui, deoarece ele vor moşteni noile însuşiri.

Pînă acum toate metodele au fost declarate static, aceasta înseamnă că adresele lor de pornire sînt stabilite rigid în momentul compilării. Cu acest lucru va trebui să ne mulţumim deocamdată. Ceea ce se poate face cu metodele virtuale vom afla în cursul următor. În acest loc vom arunca o privire asupra structurilor dinamice, deoarece programarea orientată obiect este posibilă şi în acest mod.

Cele mai agreabile structuri de date sînt statice. Ele sînt declarate, compilatorul rezervă spaţiul necesar pentru ele şi pot fi accesate în orice moment. Cu toate acestea există şi cîteva dezavantaje, care nu sînt de prim-plan, dar care în anumite circumstanţe pot avea o acţiune distrugătoare. Aşa ar fi, de exemplu, spaţiul de memorie rezervat variabilelor statice. Pentru ele stă la dispoziţie doar un segment de date - deci 64 KBytes. Acest spaţiu trebuie să fie suficient pentru toate variabilele. Dacă s-a definit un tablou (o matrice), atunci numărul elementelor este precis determinat. Dar ce se întîmplă

```
unit Str_Rout;

Interface

uses crt;
const
    SegBWS = $B800; {color}
  {SegBWS = $B000; mono}

Type

  BSPtr = ^BS;

  BS = Object
     linia,coloana : byte;
     procedure poz_cursor(X,Y : byte);
     procedure citeste_cuvint
                    (var st : string);
     procedure scrie_cuvint
                    (st : string);
     procedure muta_cuvint (Xd,Yd :byte;
          Nr : integer);
     function citeste_pozX : byte;
     function citeste_pozY : byte;
  end;

Implementation

procedure BS.poz_cursor (X,Y : byte);
begin
  linia := Y;
  coloana := X;
end;

procedure BS.citeste_cuvint
                  (var st : string);
var
  ch : char;
  OfsBWS : word;
  j : integer;
begin
  st := '';
  j := coloana;
  repeat
    OfsBWS := 2*(j-1)+160*(linia-1);
    ch := chr(Mem[SegBWS:OfsBWS]);
    if ((ch  '^') and (ch  chr(128)))
    then
      st := st+ch;
    inc (j);
  until ch in [
      chr(0)..'^',chr(128)..chr(255)];
end;

procedure BS.scrie_cuvint
                        (st : string);
var
  i,j : integer;
  OfsBWS : word;
begin
  j := coloana;
  for i := 1 to length (st) do begin
    OfsBWS := 2*(j-1)+160*(linia-1);
    Mem[SegBWS:OfsBWS+1] := TextAttr;
    Mem[SegBWS:OfsBWS] := Ord(st[i]);
    inc (j);
  end;
end;

function BS.citeste_pozX : byte;
begin
  citeste_pozX := coloana;
end;

function BS.citeste_pozY : byte;
begin
  citeste_pozY := linia;
end;

procedure BS.muta_cuvint
                  (Xd,Yd : byte;
                   Nr : integer);
var
  s : string;
  X,Y : byte;
  Save_Attr : byte;
  co : byte;
begin
  X := coloana;
  Y := linia;
  citeste_cuvint (s);
  Save_Attr := TextAttr;
  co := TextAttr div 16;
  TextAttr := TextAttr and co;
  scrie_cuvint (s);
  TextAttr := Save_Attr;
  poz_cursor (coloana+Xd,linia+Yd);
  scrie_cuvint (S);
  poz_cursor (X,Y);
end;

end.  {STR_ROUT.PAS Sfirsit}
```

atunci cînd numărul elementelor nu corespunde cu numărul de elemente declarat ? Dacă sînt mai puţine elemente decît s-au declarat, atunci se face risipă de memorie. Şi mai rău este însă atunci cînd trebuie stocate mai multe elemente decît a fost prevăzut. În acest caz nu există decît soluţia modificării programului sursă, recompilării şi relinkeditării. Inserarea sau ştergerea de elemente din astfel de tablouri este de asemenea dificilă, deoarece toate elementele care urmează după poziţia în care se operează trebuie mutate.

În cazul utilizării structurilor dina-

```
uses crt,str_rout;

type
  BS2 = Object (BS)
    color,background : byte;
    procedure stab_color (co : byte);
    procedure stab_fundal (bk : byte);
    procedure scrie_cuvint (st : string);
  end;

procedure BS2.stab_color (co : byte);
begin
  color := co;
end;

procedure BS2.stab_fundal (bk : byte);
begin
  background := bk;
end;

procedure BS2.scrie_cuvint
                    (st : string);
var
  Save_Attr : byte;
  i,j : integer;
  OfsBWS : word;

begin
  Save_Attr := TextAttr;
  TextColor (color);
  TextBackGround (background);
  BS.scrie_cuvint (st);
  TextAttr := Save_Attr;
end;

var A : BS2;

begin
  ClrScr;
  A.poz_cursor (10,10);
  A.stab_color (Red);
  A.stab_fundal (Black);
  A.scrie_cuvint ('Salut !');
end.

{LIST_8.PAS Sfirsit}
```

```
type

  Adresa = record
    nume,prenume : string[24];
    str,loc : string[32];
    tel : string[20];
  end;

var
  ad : Adresa;

begin
  {...}
end.

{-----------------------------------}
```

```
type

  AdrPtr = ^Adresa;

  Adresa = record
    nume,prenume : string[24];
    str,loc : string[32];
    tel : string[20];
    anterior : AdrPtr;
    urmator : AdrPtr;
  end;

var ad : AdrPtr;

{LIST_9.PAS Sfirsit}
```

mice nu vă loviţi de nici unul din aceste inconveniente. În schimb va trebui să vă osteniţi mai mult în timpul programării, deorece în acest caz dvs. sînteţi răspunzător pentru variabilele declarate. Avantajele care rezultă însă merită osteneala. Puteţi rezerva şi elibera spaţiu de memorie după dorinţă şi în funcţie de necesităţile curente. Înafară de aceasta, pentru declararea variabilelor dinamice, vă stă la dispoziţie întregul "heap". Adică toată memoria de lucru care nu este ocupată de sistemul de operare, de programe şi de datele lor şi de segmente de stivă. Observaţi cu atenţie scurtele secvenţe de programe din fişierul LIST_9.PAS (nu încercaţi să-l rulaţi, nu va funcţiona !). Partea de sus prezintă declaraţiile statice. Pentru variabila "ad" compilatorul rezervă 137 bytes de memorie (segment de date). În exemplul de jos este prezentată o declaraţie dinamică. Cu toate că declararea articolului este mai complexă, pentru variabila "ad" vor fi rezervaţi doar 4 bytes pentru preluarea unei adrese. Abia mai tîrziu, cînd variabila "ad" este necesară, cu ajutorul procedurii "new" se rezervă spaţiul de memorie necesar. Cu procedura "dispose" acest spaţiu de memorie poate fi eliberat din nou, după ce prelucrarea s-a terminat. Observînd cu atenţie definiţia articolului, veţi remarca prezenţa variabilelor "urmator" şi "anterior". În interiorul acestor structuri de date pot fi definiţi pointeri care să indice alte elemente de acelaşi tip. Avantajul este evident: puteţi trata un întreg lanţ de elemente de tipul "adresa". Pointerii spre articolul anterior şi următor sînt conţinuţi în articol. O inserare a unui nou element sau o ştergere a unui element influenţează doar vecinii din lanţ.

```
uses crt,str_rout;

type
  BS3 = ^BS2;

  BS2 = Object (BS)
    color,background : byte;
    procedure stab_color (co : byte);
    procedure stab_fundal (bk : byte);
    procedure scrie_cuvint
                        (st : string);
  end;

procedure BS2.stab_color (co : byte);
begin
  color := co;
end;

procedure BS2.stab_fundal (bk : byte);
begin
  background := bk;
end;

procedure BS2.scrie_cuvint
                    (st : string);
var
  Save_Attr : byte;
  i,j : integer;
  OfsBWS : word;

begin
  Save_Attr := TextAttr;
  TextColor (color);
  TextBackGround (background);
  BS.scrie_cuvint (st);
  TextAttr := Save_Attr;
end;

var A : BS3;

begin
  ClrScr;
  New(A);
  A^.poz_cursor (10,10);
  A^.stab_color (Red);
  A^.stab_fundal (Black);
  A^.scrie_cuvint ('Salut !');
  Dispose(A);
end.

{LIST_10.PAS Sfirsit}
```

```
uses crt;

type
  AdrPtr = ^Adresa;

  Adresa = record
    nume,prenume : string[24];
    str,loc : string[32];
    anterior : AdrPtr;
    urmator : AdrPtr;
  end;

  Trateaza_adresa = object
    inceput,sfirsit,ultim : AdrPtr;
    procedure creaza;
    procedure introduce;
    procedure afiseaza;
    procedure sterge;
  end;

procedure Trateaza_adresa.creaza;
begin
  inceput := nil;
  ultim := nil;
  sfirsit := nil;
end;

procedure Trateaza_adresa.introduce;
var
  A,B : AdrPtr;
  ch : char;
  i : integer;
begin
  ClrScr;
  WriteLn
      ('  Introducere adrese noi  ');
  WriteLn;
  WriteLn ('Nume        : ');
  WriteLn ('Prenume     : ');
  WriteLn ('Strada      : ');
  WriteLn ('Localitatea : ');
  repeat
    repeat
      New (A);
      if inceput = nil then
                      inceput := A;
      sfirsit := A;
      GotoXY (15,3);
      ReadLn (A^.nume);
      GotoXY (15,4);
      ReadLn (A^.prenume);
      GotoXY (15,5);
      ReadLn (A^.str);
      GotoXY (15,6);
      ReadLn (A^.loc);
      A^.anterior := nil;
      A^.urmator := nil;
      WriteLn;
      Write ('Totul este O.K. ? D/N  ');
      repeat
```

```
        ch := UpCase (Readkey);
      until ch in ['D','N'];
      for i := 3 to 6 do
      begin
        GotoXY (15,i);
        Write ('
');
      end;
    until ch in ['D'];
    A^.anterior := ultim;
    if ultim  nil then
    begin
      B := A^.anterior;
      B^.urmator := A;
    end;
    ultim := A;
    ch := ' ';
    GotoXY (1,8);
    Write ('Continuati ?  D/N    ');
    repeat
      ch := UpCase (Readkey);
    until ch in ['D','N'];
```

until ch in Ă'N'Î;

```
end;

procedure Trateaza_adresa.afiseaza;
var
  A : AdrPtr;
  ch : char;
begin
  ClrScr;
  WriteLn
    ('  Afisarea datelor introduse');
  WriteLn;
  A := inceput;
  while A  nil do
  begin
    WriteLn ('Nume       : '+A^.nume);
    WriteLn ('Prenume    : ',+A^.prenume);
    WriteLn ('Strada     : ',+A^.str);
    WriteLn ('Localitatea : ',+A^.loc);
    WriteLn;
    A := A^.urmator;
    ch := Readkey;
  end;
end;

procedure Trateaza_adresa.sterge;
var A : AdrPtr;
begin
  while inceput  nil do
  begin
    A := inceput^.urmator;
    Dispose (inceput);
    inceput := A;
  end;
end;

var adr1 : Trateaza_adresa;

begin
  adr1.creaza;
  adr1.introduce;
  adr1.afiseaza;
  ClrScr;
  WriteLn ('Heap inainte de stergere : '
                          ,MemAvail);
  adr1.sterge;
  WriteLn ('Heap dupa stergere       : '
                          ,MemAvail);
  ReadLn;
end.

{LIST_11.PAS Sfirsit}
```

```
type

  Trateaza_adresa2 = object
                       (Trateaza_adresa)
    procedure insereaza_una;
    procedure sterge_una;
  end;

{LIST_12.PAS Sfirsit}
```

Dar să ne întoarcem la programarea orientată obiect. Desigur că şi obiectele pot fi declarate dinamic. Acest fapt prezintă imensul avantaj că abia în momentul rulării se va rezerva în memorie spaţiul necesar codului metodelor necesare. Pentru aceasta să dăm un exemplu: modificaţi fişierul LIST_8.PAS introducînd înaintea declaraţiei obiectului instrucţiunea BS3 = ^BS2. Listingul trebuie să arate ca cel din fişierul LIST_10.PAS. Simbolul "^" indică compilatorului că va fi definit un pointer, deci un obiect dinamic de tipul BS2. Variabila A va fi un obiect dinamic. În scurtul nostru exemplu ea va fi instalată cu procedura new(A) şi va fi din nou eliberată cu procedura dispose(A). Sarcinile vor fi transmise obiectului pe calea cunoscută.

Totuşi pentru a deveni cu adevărat mai "dinamici" să luăm un exemplu mai complex. Examinaţi listingul din fişierul LIST_11.PAS. Pointerul AdrPtr poate indica elemente de tipul "adresa". Acestea conţin toate datele necesare, cum ar fi numele, prenumele şi adresa. În plus sînt declarate încă două variabile care pot indica elementul anterior şi următor de acelaşi tip. Cu ajutorul acestei structuri recursive lanţul de elemente stocat în memorie poate fi parcurs de la cap la coadă sau de la coadă la cap. Pentru o mai bună înţelegere să amintim încă odată faptul că variabilele "anterior" şi "urmator" sînt pointeri. Ele nu definesc deci ceva

asemănător unui articol ci stochează doar nişte adrese. În cazul nostru ele vor trebui să conţină adresa elementului anterior şi pe cea a următorului element. În fapt "AdrPtr" ar trebui declarată abia după ce "adresa" este cunoscută. În cazul variabilelor dinamice, în mod excepţional, este posibilă o predefiniţie. Compilatorul aşteaptă o declaraţie ulterioară pentru "adresa".

Să revenim la obiectul nostru. Pe lîngă cîmpurile de date în el sînt conţinute şi metodele elementare de prelucrare a datelor. După dorinţă, pot fi adăugate şi alte metode de tratare a datelor, cum ar fi de ex. sortarea adreselor. Noi ne vom limita aici la cele patru metode: "creaza", "introduce", "afiseaza" şi "sterge".

Metoda "creaza" nu face prea multe. Ea iniţializează cei trei pointeri "inceput", "ultim" şi "sfirsit" cu valori predefinite. Întrucît încă nu sa introdus nimic toţi vor primi valoarea "nil". "nil" înseamnă "not in list" (nu este în listă). "inceput" va indica întotdeauna primul element, iar "ultim" ultimul element din lanţ. "ultim" va fi utilizat ca pointer ajutător pentru realizarea cuplărilor înainte şi înapoi. Rutina de introducere nu este elegantă, însă arată foarte clar cum poate fi programtă o astfel de înlănţuire. Este vorba mai ales de instrucţiunile de "înlănţuire" a elementelor. Încercaţi să aflaţi singuri ce se întîmplă în acest caz.

Destul de simplă este şi metoda "afiseaza". Se ia pointerul de început şi se parcurge lista pînă cînd nu mai este nici un element în memorie. Instrucţiunea A:=A^.urmator preia poziţia următoare a pointerului.

Metoda "sterge" funcţionează asemănător cu metoda "introduce". Începînd din punctul de plecare sînt eliminate toate elementele din memorie. Pentru a demonstra faptul că memoria este eliberată în mod real, se afişează memoria disponibilă (heap) cu ajutorul funcţiei "MemAvail" înainte şi după ştergere. Se poate observa că după ştergere memoria disponibilă este mai mare.

Exemplul din fişierul LIST_11.PAS este foarte simplu. Obiectul însuşi este static, aceasta însemnînd că el se află de la început în memorie. Dinamică este doar structura datelor. Cu totul mulţumiţi nu putem fi însă doar cu atît. Lipseşte o mulţime întreagă de metode, cum ar fi de ex. inserarea sau ştergerea de elemente individuale, sau modalitatea de a edita un element existent. Ne putem imagina şi o metodă care să schimbe înlănţuirea elementelor astfel încît acestea să fie sortate alfabetic.

După tot ce-am aflat despre programarea orientată obiect, nu mai este o problemă declararea unui nou obiect. Să-l numim "Trateaza_Adresa2". În fişierul LIST_12.PAS puteţi observa declaraţiile necesare. Cum trebuie să arate metodele corespunzătoare vom vedea în următorul curs.

Să mai recapitulăm odată. O caracteristică foarte importantă a obiectelor este posibilitatea înlocuirii metodelor moştenite, în caz de necesitate, cu alte metode avînd acelaşi nume şi aceaşi listă de parametri. Programarea orientată obiect denumeşte această caracteristică polimorfism. Metodele înlocuite nu se pierd însă şi pot fi accesate în continuare. Turbo - Pascal face distincţie între metodele statice şi cele virtuale. Metodele statice, a căror adresă este stabilită riguros în momentul compilării implică limitări.

Obiectele pot lucra şi cu structuri dinamice, ele însele putînd fi dinamice. Structurile dinamice încă nu există în momentul compilării. Programatorul trebuie să aibă el grijă ca acestea să fie create în timpul rulării şi apoi şterse. Manipularea structurilor şi obiectelor dinamice este mai dificilă decît a celor statice, în schimb la dispoziţia programului stă întreaga memorie disponibilă (heap) şi nu doar segmentul de date de 64 KBytes. Înafară de aceasta, variabilele, cîmpurile de date şi obiectele dinamice pot fi folosite mai flexibil.

(R. M.)

MK2.DEB - continuare din pagina 51

```
a
nop
push es
push ss
push bx
push ds
mov ax,[bx]
mov bx,cs
mov ds,bx
cmp al,53
jz  12e
cmp al,48
jz  13a
cmp al,50
jz  141
cmp al,42
jz  164
mov ax,0
int 33
mov ax,a
mov bx,0
mov cx,77ff
mov dx,7700
int 33
mov ax,1
int 33
pop ds
pop bx
pop ss
pop es
nop
retf
nop
mov ax,2
int 33
jmp 133
mov ax,3
int 33
shr cx,1
shr cx,1
shr cx,1
inc cl
mov al,bl
or  al,30
pop ds
pop bx
mov [bx],cl
mov cl,3
shr dx,cl
inc dl
mov [bx+1],dl
mov [bx+2],al
jmp 135
xor bh,bh
mov bl,ah
and bl,cf
mov ax,5
int 33
shr cx,1
shr cx,1
shr cx,1
inc cl
mov ah,al
mov al,bl
or  ax,3030
pop ds
pop bx
mov [bx],cl
mov cl,3
shr dx,cl
inc dl
mov [bx+1],dl
mov [bx+2],ax
jmp 135
```

```
nmk2.bin
r cx
9f
w
q
```

MK2.DEB Sfirsit

Algoritmul de "propagare înapoi"

Reţele neuronale

Partea a II-a

Cu acest algoritm şi cu extinderea modelelor de reţea cu straturi intermediare, în ultimii ani s-au obţinut progrese importante În domeniul reţelelor neuronale.

Cuprinsul cursului:

- Partea 1: Privire de ansamblu asupra temei
- **Partea 2: Teoria algoritmului propagării înapoi**
- Partea 3: Sisteme expert şi asociativitate
- Partea 4: Recunoaşterea scrisului

Înainte de a prezenta şi de a putea înţelege algoritmul de învăţare prin propagare înapoi şi arhitectura de reţea care îi corespunde, introducem cîteva noţiuni care sînt mai intuitive pe modelul mai simplu de memorie matriceală lineară.

1.Memorie matriceală lineară

Să presupunem că juniorul d-voastră oscilează la şcoală, la două materii, între notele 4 şi 5. Dacă ambele note tind spre 4 (sau mai rău), atunci acest lucru ar fi fatal pentru atingerea scopului propus (promovarea clasei). Să mai presupunem că aţi vrea să automatizaţi procesul de supraveghere al urmaşului d-voastră (cît sens ar avea acest lucru, să rămînă deocamdată o întrebare nepusă). Atunci trebuie scris un program care să pornească un bec de avertizare dacă ambele note au ajuns "4". Pe calea clasică, acest lucru nu este greu. În programul respectiv se găsesc cu siguranţă două rînduri care arată cam aşa:

```
bec := stins;
if (nota1 <5) and
   (nota2 <5)
 then bec := aprins;
```

(Ex.1)

Funcţie de gradul de prudenţă, "and" poate fi înlocuit cu "or". În cele ce urmează însă, nu dorim să lucrăm cu reguli logice, ci să transferăm sarcina unei reţele neuronale. Urmează ca reţeaua, pornind de la două fapte date (nota1 şi nota2) să ia o decizie: dacă trebuie aprins sau nu becul de avertizare.

Prima întrebare este: cum trebuie să fie construită reţeaua? Ea trebuie să prelucreze două valori de intrare (input). Vom dota în consecinţă reţeaua cu doi neuroni de intrare (celule de intrare), a căror valoare (activitate) corespunde valorii celor două note. Ca ieşire a reţelei trebuie luată o decizie asupra stării becului de avertizare. Deci avem nevoie de un neuron de ieşire (o celulă de ieşire), a cărui activitate arată "bec stins" sau "bec aprins". Vom defini: "1" corespunzător lui "aprins", "0" corespunzător lui "stins".

Pentru ca reţeaua să poată ajunge la o decizie, deci la o valoare de 0 sau 1, neutronul de ieşire trebuie să fie conectat cu neuronii de intrare. Conectarea va consta în cazul nostru în a da valori celor două activităţi de input (se mai vorbeşte şi de ponderare) şi de a calcula rezultatul acestor ponderări. Dacă legătura (ponderea, sinapsa) între prima celulă de intrare I(1) şi celula de ieşire E o notăm cu

$$g(1,1) = g(I(1),E)$$

iar ponderea dintre a doua celulă de intrare I(2) şi celula de ieşire E cu

$$g(2,1) = g(I(2),E)$$

atunci pentru memoria matriceală lineară este valabilă următoarea formulă de calcul:

activitatea celulei de ieşire E = nota1 * g(1,1) + nota2 * g(2,1)

(Ex. 2)

Este vorba, în acest caz, de o banală înmulţire matriceală. Perechea de intrare (nota1, nota2) se interpretează ca vector de intrare de dimensiune 2 şi se înmulţeşte cu matricea de ponderi, de tip (2 x 1).

Activitatea celulei de ieşire - şi prin ea, decizia asupra activităţii becului de avertizare - este determinată de ponderile g (ce trebuiesc încă stabilite).

Concluzionînd, constatăm că am luat act de noţiuni ca celule de intrare, de ieşire, ponderi şi activitate neuronală. Cu ajutorul lor şi cu fig. 1, prima întrebare (cum trebuie construită reţeaua ?) şi-a găsit răspuns.

Cum se stabilesc acum ponderile, pentru ca reţeaua să ia decizii corecte? Dacă fixăm de exemplu g(1,1) = 2/5 şi g(2,1) = -1/5, atunci pentru perechea (nota1, nota2) = (5,5) va genera, conform Ex. 2, valoarea

E = 5 * (-1/5) + 5 * (2/5) =-1 + 2 = 1

Becul se aprinde, deşi de fapt nu există nici un pericol! Putem înţelege acum necesitatea unei formule care, pentru o problemă determinată şi căreia i s-a găsit o reţea potrivită, să calculeze ponderile corecte.

Această fază este denumită faza de învăţare a reţelei neuronale. La marea majoritate a reţelelor însă,

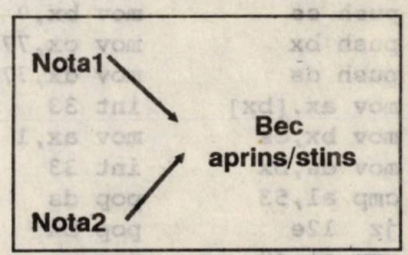


Figura 1. O reţea simplă conectată linear, cu două valori de intrare şi o ieşire.

ponderările nu pot fi calculate direct, ci numai iterativ, pe paşi, modificîndu-le de la un pas la altul în aşa fel încît eroarea globală rezultantă să fie minimă.

Este evident faptul că primul lucru care trebuie arătat reţelei sînt cazurile extreme: cînd trebuie să se aprindă becul şi cînd trebuie să nu se aprindă becul. Aceste "exemple" formează cadrul-limită pentru modul de învăţare. Vorbim de aceea de "reţele supervizate".

În exemplul de mai sus vom da ca modele de învăţare (pentru început) două condiţii:

dacă nota1 = 4 şi nota2 = 4 atunci
bec = aprins
dacă nota1 = 5 şi nota2 = 5 atunci
bec = stins

Acestea nu sînt decît două cazuri speciale ale secvenţei de program din exemplul 1. Altfel formulat, acest lucru se scrie în felul următor:

I(1) = 4 şi I(2) = 4 → E = 1
I(1) = 5 şi I(2) = 5 → E = 0

În acest mod se poate marca noţiunea de "pereche mostră de antrenament": Mostrei de intrare (vectorului de intrare) (4,4) îi corespunde vectorul de ieşire (1) iar vectorului de intrare (5,5) îi corespunde (i se asociază) vectorul de ieşire (0).

Perechile (vectori de intrare, vector corespunzător de ieşire) formează astfel o mostră de antrenament pentru faza de învăţare. Dimensiunile vectorilor sînt corelaţi cu structura reţelei: în cazul de faţă vectorii de intrare de dimensiune 2 corespund celor doi neutroni de intrare iar vectorul de ieşire de dimensiune 1 corespunde unei celule de ieşire. Numărul mostrelor de antrenament nu este limitat. Este evident însă că sarcina reţelei devine cu atît mai grea cu cît trebuie "învăţate" mai multe mostre (diferite).

Ce dificultăţi apar, poate fi evidenţiate de următoarele consideraţii: conform exemplului 2, cu cele două mostre de antrenament se obţin pentru cele două ponderi g(1,1) şi g(2,1) condiţiile

$1 = 4 * g(1,1) + 4 * g(2,1)$
şi
$0 = 5 * g(1,1) + 5 * g(2,1)$

Sau , în scriere matriceală:

$$1 = (4,4) * \begin{pmatrix} g(1,1) \\ g(2,1) \end{pmatrix} \text{ şi}$$

$$0 = (5,5) * \begin{pmatrix} g(1,1) \\ g(2,1) \end{pmatrix}$$

Rezumînd într-o unică înmulţire matriceală:

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 4,4 \\ 5,5 \end{pmatrix} * \begin{pmatrix} g(1,1) \\ g(2,1) \end{pmatrix}$$

Se obţine în felul acesta un sistem de ecuaţii lineare, pentru care există algoritmi de rezolvare cunoscuţi (de ex. Gauss). Conform rezultatelor din algebra lineară, există soluţii determinate atunci şi numai atunci cînd rangul matricii $\begin{pmatrix} 4,4 \\ 5,5 \end{pmatrix}$ este acelaşi cu rangul matricii $\begin{pmatrix} 1,4,4 \\ 0,5,5 \end{pmatrix}$. Cu alte cuvinte, acest lucru înseamnă că vectorul (0,1) poate fi scris ca o combinaţie liniară a vectorilor (4,5) şi (5,4).

Exemplele din cazul de faţă au fost alese însă în aşa fel încît acest lucru nu se întîmplă, adică nu există nici o pereche de valori pentru g(1,1) şi g(2,1) în aşa fel încît cele două ecuaţii să fie satisfăcute simultan. Dezastru: memoria matriceală liniară prevăzută pur şi simplu nu poate rezolva problema!

Probabil că în momentul acesta, stimate cititor, eşti dezamăgit: ai sperat mai mult de la reţelele neuronale! Spre consolare: nu eşti primul. Încă în 1960 Minski şi Papert au relevat astfel de slăbiciuni principiale la modelele de reţele de pe atunci.

Atunci de ce am mai tratat atît de amănunţit această memorie matriceală liniară dacă nu prea este folosibilă în practică ? Motivul este că mai întîi de toate noţiunile de bază pot fi ilustrate mai clar cu ajutorul lor, iar apoi reţelele cu propagare înapoi care urmează sînt foarte asemănătoare cu memoriile matriceale lineare, cu singura deosebire că între stratul de intrare şi cel de ieşire se mai interpune un strat intermediar (neuroni "ascunşi").

2. Reţele cu propagare înapoi

Să notăm, pentru ceea ce urmează, cu n dimensiunea vectorului de intrare şi cu l dimensiunea vectorului de ieşire. Ar exista deci n celule de intrare I(i), i = 1..n, şi l celule de ieşire, E(k), k = 1..l. Reţelele cu propagare înapoi se deosebesc prin faptul că mai au (cel puţin) încă un strat de neuroni (neuroni ascunşi), despre care putem presupune că este format din m celule, C(j), j = 1..m.

Fiecare celulă este legată cu toate celulele stratului anterior; "puterea" ei (numită şi greutate sau pondere) este reprezentată de o valoare numerică. Legătura dintre celulele I(i) şi C(j) o notăm cu g1(i,j), iar legătura dintre C(j) şi E(k) o notăm cu g2(j,k).

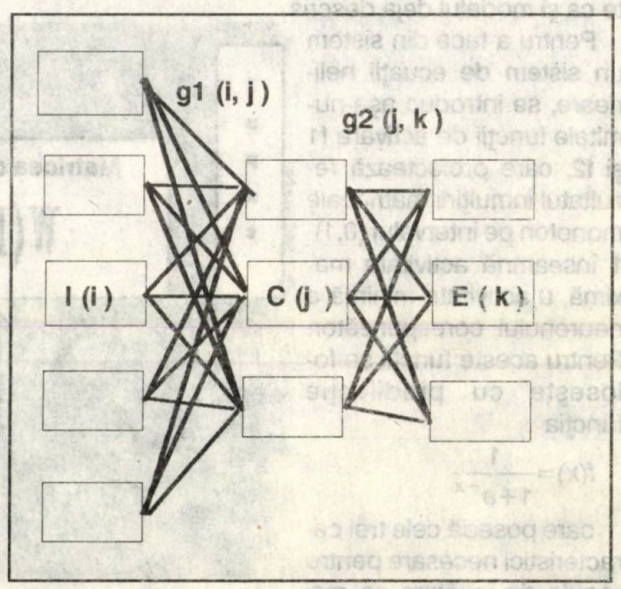


Figura 2. O reţea de propagare înapoi pe 3 nivele

$I(i) \xrightarrow{g1(i,j)} C(j) \xrightarrow{g2(j,k)} E(k)$

Pentru i = 1..n, j = 1..m şi k = 1..l rezultă atunci matricile de ponderare

W(I,H) = (g1(i,j))i,j

şi

W(H,O) = (g2(j,k))j,k

care depind de stratul de intrare şi stratul ascuns (Input şi Hidden) şi respectiv de stratul ascuns şi stratul de ieşire (Output). Cu acestea, structura reţelei de propagare înapoi este descrisă. Corespunzător exemplului 2, în acest caz are loc relaţia:

$$cp(j) = f1\left(b(j) + \sum_{i=1}^{n} g1(i,j) \times ip(i)\right) \quad j = 1..m$$

$$ep(k) = f2\left(b(k) + \sum_{j=1}^{m} g2(j,k) \times cp(j)\right) \quad k = 1..l$$

(Ex. 3)

unde ep, cp şi ip reprezintă valorile celulelor corespunzătoare (nivelul de ieşire, intermediar, intrare) pentru mostra numărul "p". Valorile b(j), b(k) sînt aşa-numite "celule de deplasament", care se tratează ca nişte ponderi normale. În lipsa funcţiilor f1 şi f2, acest model nu ar reprezenta altceva decît două memorii matriceale lineare cascadate, care ar avea aceleaşi inconveniente ca şi modelul deja descris.

Pentru a face din sistem un sistem de ecuaţii neliniare, se introduc aşa-numitele funcţii de activare f1 şi f2, care proiectează rezultatul înmulţirii matriceale monoton pe intervalul (0,1) 1 înseamnă activitate maximă, 0 activitate minimă a neuronului corespunzător. Pentru aceste funcţii se foloseşte cu predilecţie funcţia

$$f(x) = \frac{1}{1 + e^{-x}}$$

care posedă cele trei caracteristici necesare pentru regula de învăţare ce mai urmează: monotonie, nelinearitate şi diferenţiabilitate.

La memoria matriceală lineară am renunţat la prezentarea regulii de învăţare (deltarule), deoarece modelul oricum nu convenea scopurilor noastre. Algoritmul de propagare înapoi pentru reţele cu cel puţin 3 straturi este o dezvoltare a acestei reguli (generalized deltarule), care, folosit corect, rezolvă sistemul de ecuaţii în mod iterativ, prin aproximaţii.

În această fază a învăţării este important a modifica ponderile în aşa fel încît reţeaua să posede caracteristicile dorite. Pentru început, ponderilor le sînt atribuite valori mici, întîmplătoare, pentru a avea valori de pornire rezonabile. pentru o mostră p (de ex. vectorul (4,4) de mai sus); prin intermediul valorilor iniţiale prestabilite g1 şi g2 se obţin într-un "pas de prognozare" valori pentru ep. Acestea se compară cu vectorul de ieşire corespunzător vectorului de intrare (de ex. vectorul (1) din ex. de mai sus), iar diferenţa (care dă numele regulii de învăţare: deltarule) e folosită ca măsură a modificării ponderilor (un pas de învăţare).

Perfecţionarea regulii de învăţare

Modificările de ponderi sînt determinate întîi pentru matricea W(H,O). Să privim mai îndeaproape sinapsa g2(j,k) care determină puterea legăturii celei de-a "j"-a celule intermediare cu cea de-a "k"-a celulă de ieşire şi să notăm modificarea cu d2(j,k). Dacă încercăm să exprimăm acest lucru într-o formulă, obţinem următoarea expresie:

d2(j,k) = beta * (Lp(k) - ep(k))* cp(j) * f2'(x(k))

unde "beta" este o constantă de învăţare, Lp(k) este valoarea dorită (scopul învăţării) pentru mostra p la cea de-a "k"-a celulă de ieşire, ep(k) este valoarea efectiv obţinută la acea celulă de ieşire, cp(j) este valoarea celei de-a "j"-a celule intermediare la mostra p iar f2'(x(k)) este derivata funcţiei de activare f2 relativă la cea de-a "k"-a celulă de ieşire.

Pentru matricea W(I,H) rezultă următoarea prescripţie de modificare: să considerăm sinapsa g1(i,j), care reprezintă puterea legăturii dintre cea de-a "i"-a celulă

$$d1(i,j) = alfa \times ip(i) \times f1'(x(j)) \times \sum_{k-1}^{l} ((Lp(k) - ep(k)) \times f2'(x(k)) \times g2(j,k))$$

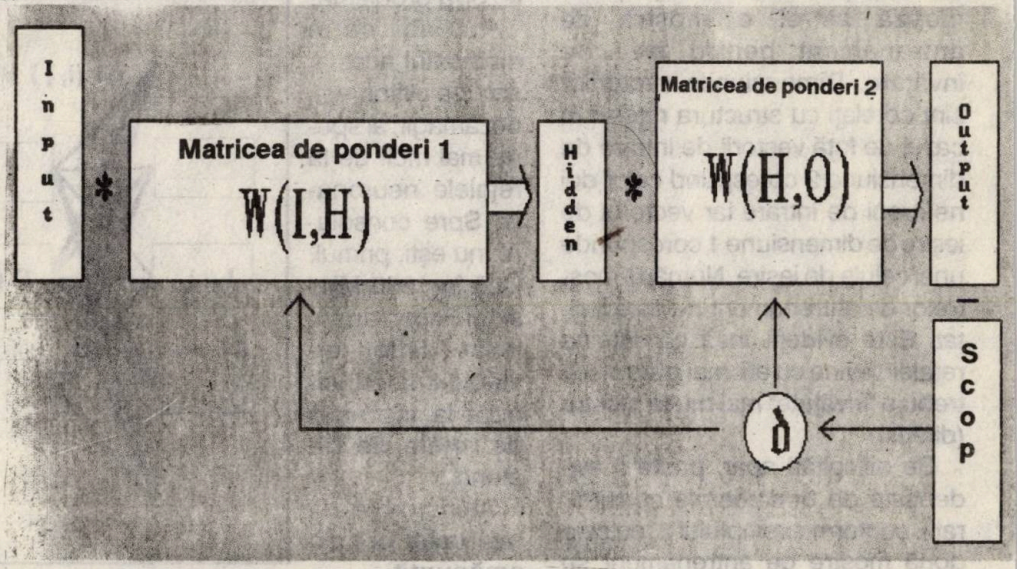


Figura 3. Vectorul de intrare determină un vector de ieşire, a cărui diferenţă faţă de vectorul scop este folosită pentru modificarea matricelor de ponderi

de intrare şi cea de-a "j"-a celulă intermediară, şi să notăm modificarea cu d1(i,j). Atunci, corespunzător acestei conexiuni, avem

unde "alfa" este tot o constantă de învăţare, Lp(k), ep(k) şi f2'(x(k)) sînt ca mai sus iar f1'(x(j)) este derivata funcţiei de activitate f1 relativ la a "j"- a celulă intermediară.

Modificarea acestor ponderi este astfel dependentă nu numai de vectorul de ieşire dorit, ci şi de ponderile plasate în urma ei. Aceste formule pot fi deduse din regula de înlănţuire pentru funcţii multidimensionale pornind de la următoarea idee:

Privind toate ponderile ca şi variabile, suma tuturor erorilor peste toate mostrele poate fi interpretată ca suprafaţă diferenţiabilă în spaţiul (n*m + m*l + 1) dimensional. Fiecare pas de învăţare îşi ia atunci "steepest descent", adică cea mai abruptă coborîre în această suprafaţă-eroare. Din punct de vedere geometric, sistemul "îşi caută un minim" în această suprafaţă.

În timpul procesului de învăţare, pentru fiecare mostră cu vectorul de ieşire corespunzător (dorit) se execută un pas de învăţare după formulele de mai sus. Dacă fiecare mostră a setului de antrenament prevăzut a fost folosit pentru cîte un pas de învăţare, o iteraţie este terminată. Algoritmul citat mai sus, după Rumelhardt, este însă destul de încet în ce priveşte timpul de convergenţă. Funcţie de problema formulată, necesită multe iteraţii. Devine evident că formularea problemelor trebuie restricţionată, iar algoritmul optimizat.

Exemplu

Destul cu teoria; să încercăm un exemplu. Problema: într-o matrice (mică) dată de pixeli, de dimensiune 5 x 5, în care fiecare pixel poate lua valorile 0 sau 1, trebuiesc recunoscute 3 modele diferite. Acestea sînt: linie verticală, constînd din 5 pixeli aflaţi unul sub celălalt, linie orizontală, constînd din 5 pixeli alăturaţi, şi linie diagonală, constînd din 5 pixeli aşezaţi diagonal.

Sarcina efectuării "manuale" a tuturor calculelor implicate poate desigur descuraja avîntul oricărui doritor de a învăţa cîte ceva despre reţelele neuronale. Pentru aceasta, vă rugăm să acceptaţi ideea urmăririi "cu ochii minţii" a experienţelor exemplificatoare pe un program de simulare a reţelelor neuronale; (un astfel de program este de exemplu NBP (Neurosoft BackPropagation) al firmei KI - Technik Schmitz din Regensburg.) (Dacă cineva din ţară cunoaşte realizări autohtone în domeniu, vom fi bucuroşi să le facem cunoscute).

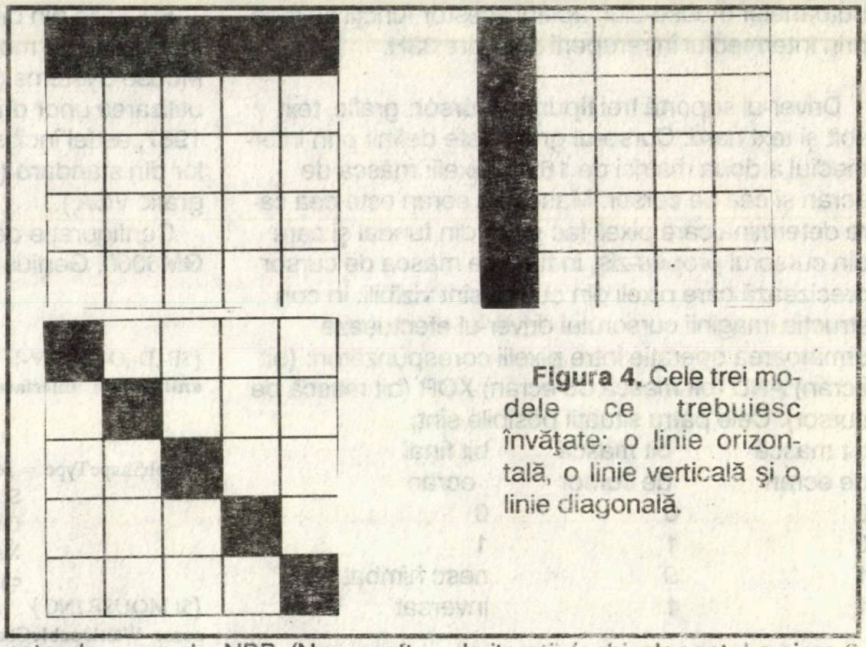


Figura 4. Cele trei modele ce trebuiesc învăţate: o linie orizontală, o linie verticală şi o linie diagonală.

Pe simulator, fenomenul poate fi urmărit şi fără o prea adîncă înţelegere a teoriei; trebuiesc exersate numai execuţia (elaborara mostrelor, verificare etc.) şi practica folosirii regulilor de învăţare (tipul, condiţiile de abandonare, numărul celulelor intermediare etc.)

În exemplul de mai sus, partea de intrare a reţelei constă din 5 * 5 = 25 neuroni de intrare. Pentru ieşire sînt prevăzuţi 3 neuroni, cîte unul pentru fiecare model (orizontală, verticală, diagonală). Mostrele de pixeli se pot edita comod cu un editor grafic oarecare.

La stabilirea mostrei de învăţare, programul întreabă care este ieşirea corespunzătoare unui anumit vector de intrare. Dacă se specifică de ex. "1" pentru orizontală, atunci de exemplu perechea (1,0,0) ca vector de ieşire împreună cu (1,1,1,1,1, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0) ca vector de intrare constituie o pereche - mostră de antrenament. După ce au fost introduse şi memorate toate cele 3 mostre, mai trebuie doar startat punctul de meniu "învăţare" şi gata - după circa 100 pînă la 150 de iteraţii (echivalentul a circa 3 pînă la 6 secunde pe un AT de 10 MHz) ponderile s-au ajustat suficient pentru ca proprietăţile dorite să se fi instalat: la mostra "linie orizontală" devine activă prima celulă de ieşire şi numai ea. La celelalte două mostre devin active a doua şi respectiv a treia celulă de ieşire.

Încă o indicaţie: la definirea reţelei cu editorul integrat de reţea trebuiesc fixate nu numai numărul de neuroni de intrare şi de ieşire, ci şi numărul de neuroni intermediari. Aici ca primă regulă se recomandă ca numărul acestora să fie cel puţin aşa de mare ca numărul neuronilor de ieşire. Pe exemplul dat, 3, astfel încît se obţine reţeaua din figura 2.

(I. F.)

Elemente de programare a mouse-ului

Pentru programatorii care n-au virat încă spre Windows prezentăm în cele ce urmează o bibliotecă şi un program Turbo Pascal cuprinzînd tehnica clasică de programare a mouse-ului sub MS-DOS. Interfaţa cu mouse-ul o constituie un driver soft ce se instalează ca un device driver (în CONFIG.SYS) sau ca un program rezident în memorie. Acesta furnizează o diversitate de funcţii pentru controlul formei şi poziţiei cursorului precum şi a stării butoanelor mouse-ului; apelul acestor funcţii se face prin intermediul întreruperii software 33H.

Driver-ul suportă trei tipuri de cursor: grafic, text soft şi text hard. Cursorul grafic este definit prin intermediul a doua matrici de 16x16 pixeli: masca de ecran şi cea de cursor. Masca de ecran este cea care determină care pixeli fac parte din fundal şi care din cursorul propriu-zis, în timp ce masca de cursor precizează care pixeli din cursor sînt vizibili. În construcţia imaginii cursorului driver-ul efectuează următoarea operaţie între pixelii corespunzători: (bit ecran) AND (bit mască de ecran) XOR (bit mască de cursor). Cele patru situaţii posibile sînt:

bit mască de ecran	bit mască de cursor	bit final ecran
0	0	0
0	1	1
1	0	nesc himbat
1	1	inversat

Odată cu forma cursorului trebuie precizat şi punctul activ al acestuia, ale cărui coordonate pe ecran devin coordonatele efective ale mouse-ului. Punctul activ al cursorului implicit (furnizat de driver) este de obicei colţul din stinga sus (0,0) .

Cursorul text soft se defineşte prin două măşti de cîte 16 biţi, cu următoarea semnificaţie:

bitul 15 - atributul de clipire (1 - blink)
12 -14 - culoarea fundalului (background)
8 -11 - culoarea caracterului (foreground)
0 - 7 - codul caracterului .

Masca de ecran determină care din atributele caracterului de pe ecran sînt menţinute. Masca de cursor determină modul în care atributele sînt alterate pentru producerea unui nou cursor. Cursorul rezultă după următoarea operaţie:

(atribute caracter) AND (mască de ecran) XOR (mască de cursor)

Cursorul text hard este cursorul clipitor standard al cartelei grafice, compus din una sau mai multe linii ; se defineşte precizînd prima şi ultima linie, între 0 şi 7 sau 0 şi 11, funcţie de tipul de display şi cartelă grafică.

Cînd o funcţie referă un cursor text sînt returnate coordonatele caracterului aflat sub cursor.

Funcţiile din bibliotecă sînt definite pentru ambele standarde de mouse: Microsoft (2 butoane) şi Mouse Systems (3 butoane). Este recomandabilă utilizarea unor driver-e de mouse mai recente decît 1987, astfel încît să se conformeze tuturor prevederilor din standard (inclusiv recunoaşterea modului grafic VGA).

Configuraţie de test: AT cu VGA, Genius Mouse GM6000, Genius Mouse Driver, Turbo Pascal 5.5 .

(ing. Alin Flaidăr)

```
{$B-,D-,O-,R-,S-,V-}
unit Mouse;  interface

type
GraphShapeType = record
                 ScreenMask : array[1..16] of word; {masca ecran}
                 CursorMask : array[1..16] of word; {masca cursor}
                 XActive, YActive: integer {coordonate punct activ}
                 end;
{$I MOUSE.INC }
const  WaitDoubleClick = 300;
                 { milisecunde intre doua clicuri pe un buton }
procedure   ClickNumber
                 ( var buttons: integer;
                   var clicks, x,y : integer);
procedure   DefEvent
                 ( Mask: integer;
                   handler: pointer );
procedure   DefGraphLocator
                 ( var shape: GraphShapeType );
procedure   DefSensitivity
                 ( deltax, deltay: integer );
procedure   DefTextLocator
                 ( StyleFlag: boolean;
                   ScrMask,CursMask: word);
procedure   DefXRange
                 ( XMin, XMax: integer);
procedure   DefYRange
                 ( YMin, YMax: integer);
procedure   GetButtonPress
                 (var button:  integer;
                  var N,x,y:    integer);
procedure   GetButtonRelease
                 (var button:  integer;
                  var N,x,y:    integer);
```

```
procedure   GetMotion
                ( var DeltaX, DeltaY: integer);
procedure   GetMouse
                (var button:  integer;
                var x,y:      integer);
procedure   HideLocator;
procedure   LightPenOff ;
procedure   LightPenOn ;
procedure   ResetMouse
                (var FoundMouse:   boolean;
                 var buttons:      integer);
procedure   RestoreMouseState
                ( buffer: pointer );
procedure   SaveMouseState
                ( buffer: pointer );
procedure   SetDoubleSpeed
                ( speed: integer);
procedure   SetMouse
                (x,y:   integer);
procedure   SetSpecialArea
                ( area: pointer );
procedure   ShowLocator;
procedure   SwapEvent
                ( var Mask, oldMask:    integer;
                  handler:              pointer ;
                  var OldHandler:       pointer );
function    MouseStateSize:   integer;

implementation

uses Dos, Crt;
type
PtrType =   record
                ofs, seg: word
            end;
var regs:   Registers;
FoundMouse:boolean;

{ urmatoarele rutine acceseaza driver-ul de mouse via Int 33H :}

procedure ResetMouse
            (var FoundMouse: boolean;
             var buttons: integer);
        { procedura de identificare a mouse-ului ; - aceasta proce-
          dura trebuie apelata inaintea oricarui alt apel de functie
          mouse, pentru a evita efecte neprevazute;
          FoundMouse = true - daca este instalat un mouse
          buttons = numarul de butoane (tip Microsoft sau Mouse
          Systems) }
      begin
            regs.ax:= 0;
            Intr($33,regs);
            buttons := regs.bx;
            FoundMouse := ( not regs.ax = 0 );
      end;

procedure ShowLocator;
        { afisare cursor text sau grafic }
      begin
            regs.ax:= 1;
            Intr($33,regs)
      end;

procedure HideLocator;
        { Ascunde cursor - aceasta functie trebuie totdeauna apela-
          ta inaintea unei opertatiuni de scriere pe ecran susceptibi-
          la sa suprascrie cursorul; - chiar in absenta cursorului dri-
          ver-ul urmareste miscarea fizica a mouse-ului ; - a nu se ui-
          ta reafisarea cursorului la momentul oportun }
      begin
            regs.ax := 2;
            intr($33,regs)
      end;

procedure GetMouse
            (var button: integer;
             var x,y: integer);
        {  returneaza starea mouse-ului: ultimul buton apasat (de la
           precedentul apel) si coordonatele mouse-ului:
           button =        0 (nici un buton apasat),
                           1 (sting),
                           2 (drept),
                           4(central)
           x,y - coordontele orizontala, verticala ale cursorului}
      begin
            regs.ax := 3;
            intr($33,regs);
            with regs do begin
                  button := bx ;
                  x := cx;
                  y := dx
            end
      end;

procedure SetMouse
            (x,y: integer);

        {  plaseaza cursorul la x,y care trebuie sa se gaseasca in do-
           meniul permis; eventual le rotunjeste la cea mai apropiata
           valoare}
      begin
            with regs do begin
                  ax := 4;
                  cx := x;
                  dx := y
            end;
            intr( $33,regs )
      end;

procedure GetButtonPress
                  (var button: integer;
                   var N,x,y: integer);
        {  N - returneaza numrul de apasari ale butonului "button"
           survenite dupa precedentul apel;
           button = 1 (sting),
                    2(drept),
                    4(central);
           x,y - coordonatele la ultima apasare}
      begin
            with regs do begin
                  ax:= 5;
                  bx := button ;
                  intr( $33, regs );
                  button := ax;
                  N:= bx;
                  x := cx;
                  y := dx
            end
      end;

procedure GetButtonRelease
            (var button: integer;
             var N,x,y: integer);
```

```
        { N - returneaza numrul de apasari ale butonului "button"
          survenite dupa precedentul apel;
          button = 1 (sting), 2 (drept), 4 (central) ;
          x,y - coordonatele la ultimul apel}
      begin
            with regs do begin
                  ax := 6;
                  bx := button ;
                  intr( $33,regs );
                  button := ax ;
                  N := bx;
                  x := cx;
                  y := dx
            end
      end;

procedure ClickNumber
            ( var buttons: integer;
              var clicks, x,y : integer);
        { returneaza numarul de clicuri (de la ultimul apel) pe buto-
          nul "buttons"
          clicks = 0 - nici un clic; = 1 - clic ; = 2 - dublu clic }
      begin
            GetMouse( buttons, x, y);
            if buttons = 1
                  then begin
                        delay(WaitDoubleClick);
                        buttons := 0;
                        GetButtonPress( buttons,clicks,x,y);
                        end
                  else clicks := 0
      end; { ClickNumber }


procedure DefXRange
            ( XMin, XMax: integer);
        { miscarea cursorului pe orizontala este limitata intre
          XMin si XMax ; daca cursorul se gaseste in afara acestei
          zone cind se efectueaza un apel cursorul este mutat in inte-
          riorul zonei. }
      begin
            with regs do begin
                  ax := 7;
                  cx := XMin;
                  dx := XMax
            end;
            intr( $33, regs )
      end;

procedure DefYRange
            ( YMin, YMax: integer);
        { miscarea cursorului pe verticala este limitata intre YMin
          si YMax; daca cursorul se gaseste in afara acestei zone
          cind se efectueaza un apel cursorul este mutat in inte-
          riorul zonei. }
      begin
            with regs do begin
                  ax := 8;
                  cx := YMin;
                  dx := YMax
            end;
            intr( $33, regs )
      end;

procedure DefGraphLocator
            ( var shape: GraphShapeType );
        { Defineste cursorul grafic conform valorilor din argumen-
          tul shape }
      begin
            with regs, shape do begin
                  ax := 9;
                  bx := XActive;
                  cx := YActive;
                  dx := ofs(ScreenMask);
                  es := seg(CursorMask);
            end;
            intr( $33, regs )
      end;

procedure DefTextLocator
            ( StyleFlag: boolean;
              ScrMask,CursMask: word);
        { Defineste cursorul in mod text;
          StyleFlag = true - cursor soft definit de ScrMask,Cursmask
          = false- cursor hard definit prin prima linie (ScrMask) si
          ultima linie (CursMask) }
      begin
            with regs do begin
                  ax := 10;
                  if StyleFlag
                        then bx := 0
                        else bx := 1 ;
                  cx := ScrMask;
                  dx := CursMask;
            end;
            intr( $33, regs )
      end;

procedure GetMotion
            ( var DeltaX, DeltaY: integer);
        { DeltaX,DeltaY - miscarea relativa a mouse-ului dupa pre-
          cedentul apel ;DeltaX 0 - mickeys spre dreaptaDeltaY 0
          mickeys in jos ; 1 mickey = inch/ 200 }

      begin
            with regs do begin
                  ax := 11;
                  intr( $33, regs );
                  DeltaX := cx;
                  DeltaY := dx
            end
      end;

procedure LightPenOn ;
      { testata numai pentru Genius Mouse Driver }
        { mouse-ul emuleaza un creion optic; apeluri de functii de
          creion optic returneaza coordontele la ultima "scriere" sta-
          tus: toate butoanele eliberate - creion ridicat;toate butoa-
          nele apasate - creion jos ("scriere") }
      begin
            regs.ax := 13;
            intr( $33, regs)
      end;

procedure LightPenOff ;

        { testata numai pentru Genius Mouse Driver } { dezactivea-
          za orice emulare de creion optic; dupa dezactivare orice
          apel de functii de creion optic vor referi doar un creion op-
          tic real }
      begin
            regs.ax := 14 ;
            intr( $33, regs )
```

```
    end;
procedure DefEvent
            ( Mask: integer;
              handler: pointer );
        { defineste o functie (handle) care se executa la aparitia
          unui eveniment hard definit de urmatoarea masca:
          Mask = 0 ( locatie de cursor modificata ),
          1 ( butonul sting apasat ),2 ( butonul sting eliberat),
          3 ( butonul drept apasat ),4 ( butonul drept eliberat ),
          5 ( butonul central apasat ),6 (butonul central eliberat),
          7 - 15 neutilizate ;-functia este referita prin pointer la ince-
          putul codului si trebuie sa indeplineasca toate conditiile
          unei intreruperi, printre care cea mai importanta este sa
          evite apelul de functii DOS, intrucit DOS-ul nu este rean-
          trant ; -la aparitia evenimentului definit de Mask progra-
          mul este intrerupt si va fi reluat din acelasi punct dupa
          executarea functiei }
    begin
          with regs do begin
                ax := 12;
                cx := $01 shl Mask ;
                dx := PtrType( handler ).ofs;
                es := PtrType( handler ).seg;
          end;
          intr( $33, regs )
    end;

procedure SwapEvent
            ( var Mask, oldMask: integer;
              handler: pointer;
              var OldHandler:  pointer );
        {  schimba functia si masca precedenta cu o functie noua ac-
          tivata de o masca noua }
    begin
          with regs do begin
                ax := 20;
                cx := $01 shl Mask;
                dx := ofs( handler^ );
                es := seg( handler^ );
                intr( $33, regs );
                OldMask := cx ;
                PtrType( handler ).ofs := dx;
                PtrType( handler ).seg := es
          end
    end;

procedure DefSensitivity
            ( deltax, deltay: integer );
        { defineste sensibilitatea mouse-ului ( in mickeys/ 8 pixe-
          li):dx - orizontal: inch/200 pe 8 pixeli: implicit: 8
           dy - vertical:------ " -------implicit: 16 ; }
    begin
          with regs do begin
                ax := 15;
                cx := deltax;
                dx := deltay
          end;
          intr( $33, regs)
    end;

procedure SetSpecialArea
            ( area: pointer );
        { dezactiveaza cursorul  la intrarea in acesta zona; la parasi-
          rea zonei cursorul trebuie reafisat; zona este definita ca
          pointer la o variabila de tipul record LeftX,Up-
          perY,RightX,LowerY: integer end}
    begin
          with regs do begin
                ax := 16;
                es := PtrType( area ).seg;
                dx := PtrType( area ).ofs
          end;
          intr( $33, regs )
    end;

procedure SetDoubleSpeed
            ( speed: integer);
        { seteaza limita de dublare a vitezei de miscare a cursoru-
          lui:daca mouse-ul se misca mai rapid de "speed"  inch/200
          pe secunda, miscarea cursorului se dubleaza; implicit
          speed = 64 ;pentru a e dezactiva dublarea vitezei se setea-
          za speed = 32767 }
    begin
          with regs do begin
                ax := 19;
                dx := speed
          end;
          intr( $33, regs )
    end;

function MouseStateSize: integer;
        { returneaza dimensiunea buffer-uluri  necesar pentru salva-
          rea starii curente a mouse-ului }
    begin
          regs.ax := 21;
          intr( $33, regs );
          MouseStateSize := regs.bx
    end;

procedure SaveMouseState
            ( buffer: pointer );
        { functie utilizata la intreruperea programului si executarea
          unui alt program care  deasemenea utilizeaza mouse-ul ; va-
          riabilei buffer trebuie sa i se aloce minimum cantitatea de
          memorie returnata de functia precedenta}

    begin
          with regs do begin
                ax := 22;
                es := PtrType( buffer ).seg;
                dx := PtrType( buffer ).ofs
          end;
          intr( $33, regs )
    end;

procedure RestoreMouseState
            ( buffer: pointer );
        { restaurarea starii anterioare a mouse-ului;trebuie apelata
          dupa terminarea programului  intrerupere. }
    begin
          with regs do begin
                ax := 23;
                es := PtrType( buffer ).seg;
                dx := PtrType( buffer ).ofs
          end;
          intr( $33, regs )
    end;


end.
```

```
{Mouse.inc}
{ o mica biblioteca de forme de cursor grafic:
      cruce, clepsidra, tinta, sageata }

const
cross: GraphShapeType = (ScreenMask:
      ($FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $0001,
       $FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $FEFF, $FEFF);
      CursorMask:
      ($0100, $0100, $0100, $0100, $0100, $0100, $0100, $FFFE,
       $0100, $0100, $0100, $0100, $0100, $0100, $0100,$0100 );
      XActive: 7; YActive: 7 );
clock: GraphShapeType = (ScreenMask:
      ($8001, $8001, $C003, $E007, $E007, $F00F, $F81F, $FC3F,
       $FC3F, $F81F, $F00F, $E007, $E007, $C003, $8001, $8001) ;
       CursorMask:
      ($7FFE, $6006, $2004, $1528, $13C8, $09D0, $05A0, $0240,
       $0340, $04A0, $0910, $1008, $1308, $27E4, $6FE6, $7FFE    );
      XActive: 7; YActive: 7 );

target: GraphShapeType = (ScreenMask:
      ($FFFF, $C007, $8003, $0001, $0001, $0001, $0001, $0381,
       $0381, $0381, $0001, $0001, $0001, $0001, $8003, $C007    );
      CursorMask:
      ($0000, $0100, $1FF0, $2108, $4104, $47C4, $4924, $4824,
       $FC7E, $4824, $4924, $47C4, $4104, $2108, $1FF0, $0100    );
      XActive: 7 ; YActive: 8 );
arrow: GraphShapeType = (ScreenMask:
      ($1FFF, $0FFF, $07FF, $03FF, $01FF, $00FF, $007F, $003F,
       $001F, $00FF, $007F, $203F, $E01F, $E1FF, $E3FF, $E7FF );
      CursorMask:
      ($E000, $D000, $C800, $C400, $C200, $C100, $C080, $C040,
       $C1E0, $D900, $F880, $D840, $19E0, $1A00, $1C00, $1800  );
      XActive: 1; YActive: 0 );
```

```
{$A+,B-,D+,F-,I+,L+,N+,O-,R-,S-,V-}
{$M 16384,0,655360}

program TestMouse;

uses  CRT,Graph,Mouse;


var driver,mode, buttons,clicks:integer;
      quit,found  : boolean;
      i,x,y:   integer;

const k: integer = 0;

begin
            { initializeaza  VGA: }
      driver := VGA; mode := VGAHi;
      InitGraph(driver,mode,'c:\tp');
            { pune pixeli aleatoriu pe ecran: }
      for i := 1 to 1000 do
            PutPixel
                  (random(GetMaxX),
                   random(GetMaxY),
                  random(15));
            { initializeaza mouse-ul }
       ResetMouse( Found, buttons);
      if not found then exit;
      rectangle( 0,0,200,150);
      rectangle( 300,100, 480,450);
            { defineste primul cursor: }
       DefGraphLocator( arrow);
            { plaseaza mouse-ul : }
      SetMouse(250,150);
            { afiseaza cursorul: }
      ShowLocator;
            { mareste sensibilitatea pe orizontala la
                  16 inch/200 / 8 pixeli }
      DefSensitivity( 16,16);
      repeat
            buttons := 1;
            ClickNumber( buttons, clicks, x, y );
            case clicks of
            2:       begin
                  { ascunde cursorul inaintea unei operatii de desen: }
                  HideLocator;
                  rectangle( x,y, x+30, y+30);
                        { reafiseaza cursorul :}
                  ShowLocator;
                  delay(10)
                  end;
            1:       begin
                  HideLocator;
                  circle(x,y,30);
                  ShowLocator
                  end
            end;
            if (x < 200 ) and (y < 150)
                  then begin
                        HideLocator;
                        DefGraphLocator( target);
                        ShowLocator ;
                        repeat
                              GetMouse(buttons,x,y)
                        until (x > 200) or (y > 150);
                        HideLocator;
                        DefGraphLocator( arrow);
                        ShowLocator
                        end
                        else if (x > 300) and (x < 480)
                                    and (y > 100) and (y < 450)
                         then begin
                              HideLocator;
                              DefGraphLocator( clock);
                              ShowLocator;
                              repeat
                                    GetMouse( buttons,x,y)
                              until (x < 300) or (x > 480)
                                    or (y < 100) or (y > 450);
                               HideLocator;
                              DefGraphLocator( arrow);
                              ShowLocator
                        end ;
      until buttons = 2;
      HideLocator;
      CloseGraph;
end.
```

dBase cu mouse

În dBase pot fi programate foarte uşor meniuri de selecţie şi întrucît începînd din versiunea III Plus într-un program pot fi integrate şi programe scrise în limbaj de asamblare, nu există nici un motiv care să ne determine să renunţăm la a folosi un mouse dintr-un program dBase. Procedurile din listingul INIT.PRG prezintă cele mai importante funcţii mouse în mod text. Este necesar doar un driver de mouse compatibil Microsoft. Cu uşurinţă pot fi integrate şi alte funcţii. Soft-ul de driver propriu-zis este prezentat în programul în limbaj de asamblare MK2.DEB. Cu comanda "DEBUG < MK2.DEB" se obţine automat fişierul MK2.BIN.

Primii doi parametrii din INIT.PRG sînt variabile numerice şi sînt utilizaţi pentru înregistrarea numărului coloanei şi respectiv liniei. Al treilea parametru este o variabilă caracter care va conţine starea tastelor de pe mouse şi respectiv numărul tastei. Numărul "0" este folosit pentru stînga şi numărul "1" pentru dreapta. Starea "0" semnifică neapăsat, "1" apăsat stînga, "2" apăsat dreapta şi "3" apăsat atît stînga cît şi dreapta. Al patrulea parametru conţine numărul de apăsări pe taste: "3" înseamnă apăsat de trei ori.

Din EXEMPLU.PRG modul de conlucrare dintre fişierul binar şi procedurile de mouse devine clar.

(MK2.DEB în pag. 41)

(R.M.)

```
set talk off
set procedure to init
clear
do init
@ 0,14 say "Demo-selectie meniu cu ;
mouse"
do feldaus with 2,23,10,4,"Adaugare  ;
ModificareStergere  Sfirsit"
@ 10,10 say "Pentru selectie apasati ;
tasta stinga a mouse-ului"
x = 0
y = 0
b = "0"
z = "0"
inca = .T.
do while inca
do while z="0" .or. (y<4) .or. (y>7)
do button with x,y,b,z
enddo
? chr(7)
z = "0"
@ 14,16 say "Ati selectat" .
do case
case y=4
@ 14,30 say "Adaugare    "
case y=5
@ 14,30 say "Modificare "
case y=6
@ 14,30 say "Stergere    "
case y=7
@ 14,30 say "Sfirsit     "
inca = .F.
endcase
@ 14,42 say "!"
enddo
set talk on
do hide
EXEMPLU.PRG  Sfirsit
```

```
procedure init
set talk off
clear
load mk2
call mk2 with "I"
return
procedure show
call mk2 with "S"
return
procedure hide
call mk2 with "H"
return
procedure pos
parameters x,y,but
par = "P   "
call mk2 with par
x = asc(substr(par,1,1))
y = asc(substr(par,2,1))
but = substr (par,3,1)
return
procedure button
parameters x,y,but,numar
par = "B"+but+"  "
call mk2 with par
x = asc(substr(par,1,1))
y = asc(substr(par,2,1))
but = substr(par,4,1)
numar = substr(par,3,1)
return
procedure feldaus
parameters
  y1,x1,zb,zz,contin
@ y1,x1 to
  y1+zz+1,x1+zb+1
y = y1+1
x = x1+1
a = 1
do while zz>0
@ y,x say ;
  substr(contin,a,zb)
a = a+zb
y = y+1
zz = zz-1
enddo
return

INIT.PRG Sfirsit
```

Vocabularul esenţial

Este posibil să consacri PC-urilor mii de lucrări; noi nu ne propunem, în cîteva zeci de pagini, decît esenţialul.

Pentru a înţelege bine conţinutul articolelor publicate este necesar ca cititorul să se familiarizeze cu un vocabular specializat. Noi vom încerca, (măcar de acum înainte), să explicăm fiecare cuvînt nou, fiecare concept nou; fiecare prescurtare, dar înţelegerea unui vocabular minimal este necesară înainte de a plonja în lectură. Pentru ca şi cititorul neofit să ne poată înţelege fără dificultate, propunem să începem cu un "pico-dicţionar" de definiţii foarte simple ale cuvintelor a căror cunoaştere este indispensabilă, urmînd ca în numerele viitoare să-l complectăm cu alte definiţii.

Cititorul care va dori să-şi completeze vocabularul său informatic va putea consulta cu folos "Dicţionarul de informatică" apărut la Editura ştiinţifică şi enciclopedică Bucureşti, 1981.

Alimentarea Este blocul funcţional (din calculator) care generează, (pornind de regulă de la tensiunea de alimentare de 220V din reţea), diversele tensiuni necesare funcţionării diferitelor componente ale calculatorului (+5V cc, +12V cc, -12V cc, +9V cc, -5V cc etc.)

"AT" Calculator personal echipat cu un microprocesor Intel 80286. AT este prescurtarea pentru "Advanced Technology". Dacă în 1990 erau încă cele mai vîndute calculatoare personale, în momentul de faţă au ajuns în amurgul vieţii; locul lor va fi luat de calculatoare mai puternice.

Bit Unitate de măsură a informaţiei. Cifră în sistemul de numeraţie binar.

Compatibil IBM Acest termen este utilizat pentru a desemna calculatoarele care sînt capabile de a utiliza programele şi plăcile de extensie prevăzute pentru calculatoarele personale IBM ("PC", "PC-XT", "PC-AT", PC-386" sau PC-486).

Harddisk (disc dur, disc rigid) Harddisk-ul este un suport de memorie de foarte mare capacitate (cîteva sute de milioane de caractere). Spre deosebire de dischete, harddisk-urile nu sînt amovibile. Bineînţeles, dacă nu mai dorim să păstrăm o informaţie pe harddisk, este foarte simplu s-o ştergem.

Dischetă Discheta este suportul, cel mai răspîndit, de memorie externă pentru PC-uri.

Dischetele se inserează în lectorul de dischete care poate fie să scrie fie să citească informaţii de pe ele. Dischetele sînt amovibile şi conţinutul lor poate fi astfel citit de toate calculatoarele "compatibile".

Hardware (hard) Totalitatea "fierăraielor" unui sistem de calcul, deci a circuitelor, a dispozitivelor şi a echipamentelor.

IBM International Business Machines. Cel mai mare producător de tehnică de calcul (S.U.A.).

Imprimantă Imprimanta este un fel de "maşină automată de scris" conectată la calculator şi, de regulă, capabilă să imprime atît texte cît şi grafică.

Laptop Calculator portabil

Memoria de masă Memoria de masă este memoria externă a calculatorului (dischetă, harddisk, disc optic, etc.). În raport cu memoria internă, această memorie este adesea de o capacitate mult mai mare, de unde şi numele care i s-a dat.

Memoria principală (sau internă) Memoria internă a calculatorului este cea care permite memorarea programelor şi efectuarea calculelor. Memoria internă a calculatoarelor sub MS-DOS este cuprinsă între 256 KBytes şi 4 GBytes.

Monitor (display) Monitorul este cel care permite citirea datelor introduse de la tastatură sau comunicate de calculator. Monitorul poate fi monocrom sau color, în tehnologie "tub cinescopic", cu "cristale lichide" sau cu "plasmă". Monitorul este "guvernat" de o placă electronică plasată în calculator. Fiecărui tip de monitor (monocrom, grafic, color, înaltă rezoluţie, etc.) îi corespunde un tip de placă electronică.

Mouse "Şoarece". Echipamentul care permite transmiterea deplasărilor relative ale mîinii, cursorului de pe monitor.

MS-DOS MS-DOS este un sistem de operare al calculatoarelor personale. Cu alte cuvinte, este programul capabil să gestioneze complet funcţionarea calculatorului. Este imposibil să utilizezi un calculator fără sistem de operare. Există mai multe versiuni corespunzînd etapelor evolutive ale echipamentelor. Denumirea vine de la Microsoft - Disk Operating System.

Octet (byte) Unitate de măsură a informaţiei. 1 octet = 8 biţi. Multiplii: Ko (Kb, Kilooctet, Kilobyte, Kbyte) 1 Ko = 1024 octeţi (2^{10}); Mo (Mb, Megaoctet, Megabyte, Mbyte) 1 Mo = 1024 Ko; Go (Gb, Gigaoctet, Gigabyte, Gbyte) 1 Go = 1024 Mo.

"PC" Personal Computer sau calculator personal. Acesta este termenul pe care-l utilizăm pentru a desemna calculatoarele personale IBM sau compatibile, şi aceasta fără a face distincţie între diferitele tipuri constructive.

PC-DOS PC-DOS este numele utilizat de IBM pentru a desemna MS-DOS. PC-DOS diferă de MS-DOS prin cîteva detalii, dar acestea nu sînt semnificative.

Periferic Perifericele sînt toate aparatele care sînt conectate la unitatea centrală a calculatorului (la carcasă): tastatura, monitorul, imprimanta, etc.

Placă de extensie Aşa sînt denumite plăcile electronice care pot fi inserate în amplasamente prevăzute în acest scop în calculatoarele personale.

Fiecare placă electronică este specializată (grafică, memorie, etc.) şi prin adăugarea de plăci de extensie un PC poate fi transformat complet pentru a-l adapta nevoilor personale.

(Micro-) Procesor Procesorul este circuitul integrat ("inima") care gestionează calculatorul. Calculatoarele IBM şi "compatibile" utilizează mai multe tipuri de circuite integrate compatibile între ele: 8088 (care echipa IBM-PC -urile şi anumite compatibile), 8086 (mai rapid decît precedentul), 80186 (foarte puţin utilizat), 80286 (care echipează IBM-PC AT-urile şi compatibile), 80386 (care va fi procesorul de bază al anilor imediat următori), 80486 (extrem de puternic, dar şi foarte scump).

Sistem de operare Sistemul de operare este o colecţie de programe capabilă să gestioneze complet şi în amănunt un calculator, degrevînd utilizatorul de detalierea unor sarcini pe care le preia. (să gestioneze tastatura, să utilizeze dischete, să tipărească la inprimantă, etc).

Sistemul de operare este un veritabil dirijor al calculatorului, el ştie ce poate să facă fiecare participant, cînd trebuie să facă şi cum trebuie să facă.

Software (soft) "Inteligenţa" calculatorului., Totalitatea programelor de pe un calculator.

Streamer Streamer-ul este o unitate rapidă de bandă magnetică, folosită de regulă ptr. salvarea periodică a datelor de pe harddisk. Este capabil să copieze (în ambele sensuri) în cîteva minute cîteva sute de mii de caractere.

"XT" XT-urile sînt tot calculatoare IBM (sau compatibile) dotate cu un harddisk sau posedînd caracteristicile necesare pentru a fi dotate cu un harddisk. Denumirea vine de la eXtended Technology.

BORLAND

Anunta

Programul de AMNISTIE

"LOVE BORLAND PRODUCTS"

Daca detineti copii neinregistrate de produse program BORLAND (Paradox, Quattro Pro, Turbo C, Turbo Pascal) puteti obtine ultima versiune a produsului ORIGINAL si deveni utilizator inregistrat si LEGAL.

Pentru detalii, va rugam sa va adresati firmei

LOGIC

BUCURESTI: 75.49.00 - 75.71.35

SIBIU: (924) 4.66.52 - (924) 4.54.75

ES COM

FRUMOS & INTELIGENT

Doriţi să achiziţionaţi un calculator de marcă ?
Doriţi să nu aveţi probleme cu calculatorul D-voastră ?

Firma **Micro-ATCI** vă poate furniza AT-uri 286, 386SX, 386, 486, Laptop-uri produse de firma **ES"COM** din Germania. Orice configuraţie este posibilă. O gamă largă de periferice şi accesorii vă stă la dispoziţie.

Doriţi să realizaţi o reţea cu produsele firmei **ES"COM** ?
Un sistem configurat după dorinţă vă aşteaptă.

Deviza noastră:
UN CALCULATOR DE EXCEPŢIE LA UN PREŢ DE EXCEPŢIE !

Nu uitaţi adresa noastră:

Micro - ATCI
C.P. 64, O.P. 1
4300 Tîrgu Mureş
Tel. 954/17024

micro ATCI

Din scrisorile sosite la redacţie

"Relativ la nr. 3/90: 1.Scop/domeniu - Nu pot să nu fiu de acord: PC-uri, MS-DOS. În general, şi desigur şi-n România, calculatoarele personale se identifică cu clone IBM-PC. Îmi pare sincer rău pentru Apple, HP şi 68030, dar n-avem ce face! Dar:

1.a.Foarte bine că excludeţi din start Sinclair Spectrum şi compatibile HC, dar ce facem cu CP/M-ul pe 8 biţi? De acord că-s depăşite moral, dar sînt încă foarte răspîndite şi nu se vor arunca prea curînd. Plus că despre HC85 & Co s-a tot scris, au apărut şi cărţi, şi articole, de exemplu în Tehnium, şi înainte şi după Revoluţie, au şi reviste specifice... dar despre CP/M, nimic. Este un gol de informare ce ar trebui umplut.

1.b.La cĕlălalt capăt al gamei, ce facem cu sistemele mari, precum VAX sau SEL? Utilizatorii de supermini sînt foarte puţini la noi în ţară. Cîţi au acces la VAX? Să însemne asta că n-o să avem niciodată informaţii în romămeşte despre VMS & Co ?(...)

6. ştirile din pagina 4 sînt inedite, interesante, dar ... f. incomplete, neclare. Nu m-am lămurit de loc! În ce constă brevetul lui Hyatt ? La fel cu înlocuitorul de harddisk. Singurul lucru cert este o greşeală în text: "timpul de acces e de 100 de ori mai mare..."

1/91

1.Aţi atras atenţia f. bine cu 286/16MHz, dar n-aţi insistat suficient. NU există 80286 lucrînd la 16 MHz! Taiwanezii oferă aşa ceva, dar este un bluff. Procesoarele sînt sortate şi forţate din cuarţuri la 16 MHz în loc de 12. Se încălzesc mai tare, fiabilitatea scade considerabil. Mai ales la noi, este neplăcut să-ţi crape...Cel mai rău este că nu se asigură nici un avantaj real: restul sistemului (nu doar RAM-ul şi harddiskul, ci tot: interfeţe, coprocesoare, etc.) nu-i adaptat ca viteză, performanţele-s dictate tot de veriga cea mai slabă. Deci programele nu rulează mai repede. Ci, în unele cazuri, chiar mai lent - testele au dovedit-o. Ca să "aştepte" DRAM-ul sau alte componente, se inserează stări de WAIT ... procesarea va dura mai mult ca pe un 286 "cinstit", la 12 MHz!"

ing. Cristian Malide, Bucureşti

"Poate n-ar fi rău să se facă o analiză preţ/performanţă detaliată a calculatoarelor de provenienţă indigenă sau străină care există la momentul actual în România, nu de alta, dar tare am impresia că preţurile alea uriaşe nu au o bază reală pe care să se sprijine, în majoritatea cazurilor."

Bogdan Condurăţeanu, student, Bucureşti

"În urma citirii repetate a articolului "Cum se lucrează cu Extended Memory", îmi persistă o întrebare: totuşi, pînă la urmă, cum să procedez efectiv pentru a putea folosi Extended Memory ?? Cred că este o problemă care ar trebui lămurită pentru toţi cei care nu sînt hard-işti sau asambl-işti.

În nr. 1/90 al revistei aţi prezentat elementele limbajului de comandă BATCH, din această prezentare lipsind ASK; chiar dacă această facilitate introdusă de Norton se livrează separat fiind mai puţin utilizată, ea ar fi trebuit totuşi prezentată pentru ca prezentarea dvs. să alcătuiască un tot unitar. Scăparea dvs. m-a dus cu gîndul la acei specialişti din diferite domenii (nu este numai cazul informaticii) care doresc să obţină bani de pe urma cunoştiinţelor lor fără a dezvălui totuşi unele lucruri de amănunt în lipsa cărora toată expunerea lor anterioară este d.p.d.v. practic ineficientă sau chiar periculoasă."

ing. Cristian Ivanov, Bucureşti

"Computerul meu preferat este Macintosh al firmei Apple, firmă cu care şi corespondez. Astfel, sînt în măsură să vă semnalez cîteva erori strecurate în revistă:

Memoria lui Mac LC este de 2 Mbyte, extensibilă pînă la 10 Mbyte, nu 8 cum scrieţi dvs. Noul floppy-drive, în versiune internă sau externă, se numeşte Super-Drive, cu D majuscul. El nu citeşte dischetele doar în format MS-DOS, OS/2, PRO/DOS ci şi în format Macintosh (logic!) şi Apple II (capacităţi între 400 kbyte şi 1,4 Mbyte). În revista americană Popular Mechanics sînt date cîteva preţuri la Macintosh, şi îmi face plăcere să vi le comunic pt. a fi publicate:

Mac Classic - 999$

Mac Classic 2 Mbyte + harddisk - 1499$

Mac LC monitor 12" color - 2499$

Mac II şi monitor 13 " color - 3430$

Mac II ş monitor 13" color, varianta 5/80 - 4000$"

Ştefan Dragoş, Cluj

Redacţia îşi rezervă dreptul de a opera scurtări în text, fără a schimba sensul celor comunicate. Opiniile exprimate aparţin autorilor ; redacţia nu îşi asumă responsabilităţi asupra datelor sau opiniilor reproduse.

În numărul viitor:

- **Turbo C++**
- **DR DOS 5.0 sau MS-DOS 5.0 ?**
- **Harddisk-uri**

Mica publicitate

Vînd AT 286 cu Hercules, 1Mbyte RAM, harddisk 40 Mbyte, 380.000 lei, Cobra 10.000 lei, tel. 90/11.44.76, Bucureşti.

Doresc să **colaborez** cu posesori de **Commodore 64**. Paraschiv Teodor Cezar 63 6200 Galaţi.

Cumpăr jocuri compatibile pentru **Commodore PLUS/4**. tel. 939/22488.

Cumpăr floppy-diskuri de 5,25". Oferte la tel. 947/14362, sîmbătă şi duminică orele 11-20 resp. 9-11 - student -.

Vînd calculator compatibil IBM cu monitor color, harddisk 32 Mbyte, imprimantă. tel. 954/47165.

Important !

Revista "**if**" îşi propune editarea unui catalog al întreprinderilor (de stat sau particulare) a căror sferă de preocupări are tangenţă cu informatica (producători hard, producători soft, depanatori, distribuitori, comercianţi, etc.).
Catalogul va conţine adresele întreprinderilor, o scurtă descriere a obiectului de activitate al acestora şi o prezentare succintă a produselor şi serviciilor oferite.
Catalogul va fi distribuit prin reţeaua de difuzare a revistei "**if**", putînd ajunge astfel direct la persoanele interesate. Întreprinderile interesate de publicarea acestui catalog sînt rugate să ia legătura cu redacţia.

Important !

Căutăm difuzori/distribuitori autorizaţi pentru difuzarea revistei "**if**" !.
Condiţii avantajoase !
Rugăm persoanele interesate să ia legătura cu redacţia.

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if,** pe o perioadă de |_____| apariţii, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 57 lei. Preţul unui abonament se va calcula cu formula: 57 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

if 4 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

LOGIC

Societate comercială româno-americană

SIBIU, Calea Dumbrăvii 21,tel.(924)46652,45475 Fax:(924)46704
BUCUREŞTI, Bulevardul Tineretului 8-10,tel.(90)754900 Fax:(90)757135
Cont valută BRCE Buc.: 47.38.112.300-2
Cont lei BRCE Buc.: 30.08.112-9
Cont lei BC Sibiu: 30,80.4.03.01

LOGIC vă oferă produsele *BORLAND* în România

* PARADOX 3.5
* PARADOX SQL LINK
* PARADOX RUNTIME 3.5
* PARADOX ENGINE
* QUATTRO PRO 2.0
* QUATTRO PRO SHOW
* TURBO PASCAL PROFESSIONAL 6.XX
* TURBO PASCAL RUNTIME LIBRARY 6.XX
* TURBO PASCAL 6.XX
* TURBO C++ PROFESSIONAL
* TURBO C++ RUNTIME LIBRARY
* TURBO C++
* TURBO C 2.XX
* SIDEKICK 2.0
* TURBO DEBUGGER & TOOLS 2.XX
* SPRINT 1.XX
* SUPERKEY 1.XX

Firma **LOGIC** asigură inregistrarea licenţelor de utilizare si acordă reduceri de preţ pentru versiunile noi.

Important !

Revista "if" îşi propune editarea unui catalog al întreprinderilor (de stat sau particulare) a căror sferă de preocupări are tangenţă cu informatica (producători hard, producători soft, depanatori, distribuitori, comercianţi, etc.). Catalogul va conţine adresele întreprinderilor, o scurtă descriere a obiectului de activitate al acestora şi o prezentare succintă a produselor şi serviciilor oferite. Catalogul va fi distribuit prin reţeaua de difuzare a revistei "if", putînd ajunge astfel direct la persoanele interesate. Întreprinderile interesate de publicarea acestui catalog sînt rugate să ia legătura cu redacţia.

Important !

Căutăm difuzori/distribuitori autorizaţi pentru difuzarea revistei "if".
Condiţii avantajoase ! Rugăm persoanele interesate să ia legătura cu redacţia.

ABTECH comercializează toată gama de PC-uri (286/386/486) compatibile sută la sută IBM, precum şi reţele de calculatoare.

Comuterele **ABTECH** sînt fabricate în S.U.A., chiar de firma **ABTECH**.

Beneficiarul poate să-şi comande configuraţia PC-ului (dimensiune RAM, HDD, tip monitor şi adaptor etc.), precum şi a reţelelor.

Găsiţi la noi, **la cea mai bună calitate**:

sisteme desktop:

- 8088 10 MHz 1 Mbyte RAM
- 286-12
- 386-SX
- 386-20
- 386-25
- 386-33
- 486-25

harddisk-uri:
20..338 Mbyte

portabile:286-12

- 386SX
- 386-20
- 386-25
- 386-33

ABTECH™

ABSTRACT TECHNOLOGY S.R.L.

Bucureşti,
Sector 2, Calea Moşilor 201,
Bloc 9, Scara B., Et. 7, Apt. 53
Telefon: 10.42.73 * 10.09.65

ABTECH

if

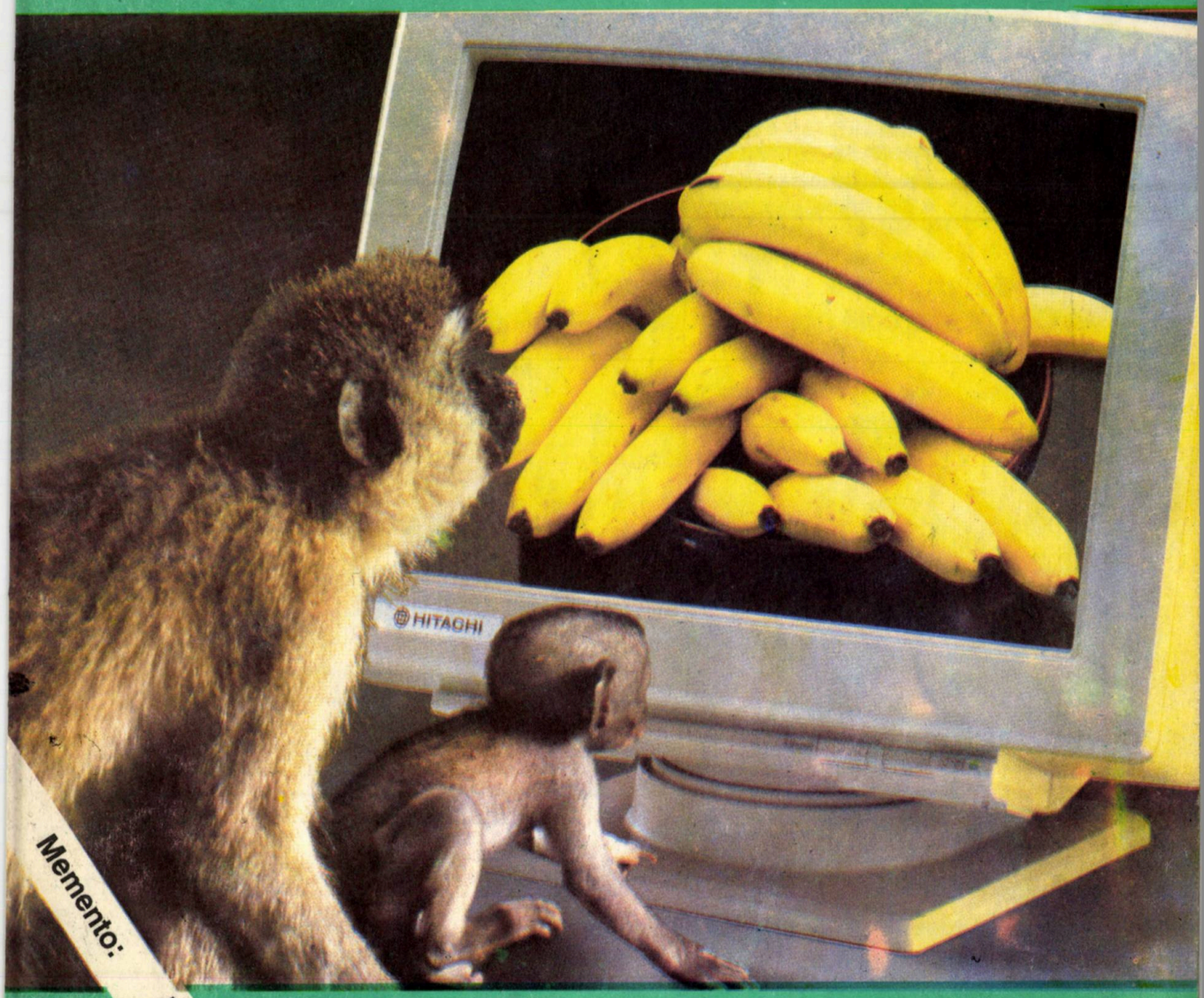
CALCULATOARE PERSONALE

5 / 91

(8)

57 Lei

Revistă lunară editată de Micro ATCI Tîrgu Mureş



Memento:

Funcţii Turbo-C

MS-DOS 5.0 vs. DR-DOS 5.0

Dosar: Turbo C++ 1.0

Paradigma Paradox

Standarde grafice:

Tiga, 8514/A, XGA

AbMod

Ábrahám & Moldovan s.n.c.

vă oferă tehnologie americană
inteligenţă europeană
preţuri orientale la

produse din domeniile:

- tehnică de calcul (hardware şi software)
- telematică şi birotică

servicii în aceleaşi domenii:

- lucrări de cercetare / proiectare
- studii de dezvoltare şi modernizare
- studii de marketing pe piaţa internă/externă
- consulting şi asistenţă tehnică
- instalare, şcolarizare, service
- sisteme "la cheie"

condiţii de livrare: 2 - 4 săptămîni

Cîteva exemple din lista noastră de preţuri:

XT

<< Preţ sistem complet 89.000 lei >>

AT-286 & AT-386

<<Preţ sistem complet [lei] > >

AT	Winchester capacitate formatată	monitor + interfaţă		
		mono 14"	EGA 14"	VGA 14"
286	40 Mb	199.000	249.000	299.000
	80 Mb	275.000	325.000	375.000
386	80 Mb	399.000		499.000
	160 Mb	549.000		649.000

Imprimante matriciale

Tipul	Lăţimea	Nr. ace	Preţ [lei]
LC 20	80 car	9 pin	49.000
LC 15	132 car	9 pin	79.000

Mouse serial compatibil Microsoft . . . . . 7.700 lei

Copiatoare FC-2 (Toshiba BD 2810) . . . . . 255.000 lei

MS-DOS v4.01 . . . . . 12.000 lei

Cei interesaţi pot obţine gratuit lista de preţuri adresîndu-se la

AbMod Magazin Hardware-Software:

b-dul 22 Decembrie nr. 135

AbMod Sediul central:

str. Moţilor nr. 91, 3400 Cluj-Napoca, tel./fax 95-111090

AbMod Sucursala Oradea:

str. Dacia nr. 40, 3700 Oradea, tel. 991-60278, 44476, fax 991-56881

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:

ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:

stud. Daniel Buleu
ing. Cristian Malide
stud. Ioan Tiberiu Socaciu

Tiparul: tipografia Tîrgu Mureş
Tipografi:

Halaţiu Mihai,
Cormoş Mircea

Revista apare lunar.

Preţ: 57 lei

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 64, O.P. 1,
tel. 954/31660 (direct),
33.612, 24.158, 20.057,
33.511, int. 134 sau 189
fax 954/35208,
telex 65354.

Manuscrise originale sau listing-uri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei.

Fantázia în primejdie

O revistă "îţi conferă puteri însemnate"(*), "dar n-ai voie să le foloseşti", "niciodată n-ai voie să intervii, indiferent de ce vei vedea, căci începînd din clipa aceasta propria ta părere nu mai are nici o importanţă". "Trebuie să laşi să se întîmple tot ce se întîmplă. Totul trebuie să aibă aceeaşi importanţă pentru tine, şi binele şi răul, şi ceea ce-i frumos, şi ceea ce-i urît, şi prostia, şi înţelepciunea". "Nu ai voie decît să cauţi şi să întrebi, nu însă să şi judeci după propria ta judecată. Să nu uiţi asta niciodată"!

Bine, dar sîntem muritori şi sîntem slabi şi nu sîntem noi aleşii. Şi uităm şi greşim!

Nu puţine publicaţii şi nu doar cele aflate "de o anumită parte a presei" au avut în ultimele zile titluri de genul: "Este ameninţată existenţa presei independente din România". Noi ce-am mai putea adăuga. Sîntem şi noi în derivă în acest triunghi al Bermudelor: hîrtie - spaţiu tipografic - difuzare. Cînd toate aceste elemente se scumpesc, în NLL (noua limbă de lemn) "se liberalizează", nu putem alege decît între a scumpi şi noi revista, sau a renunţa la editarea ei. Ne este ruşine pentru fiecare scumpire şi nu ne consolăm făcînd comparaţii de genul: de preţul unei reviste cîte felii de salam (cu sau fără soia) s-ar putea cumpăra. Oricum, "primum vivere".

O revistă trăieşte prin cititorii ei. Noi dorim ca revista să trăiască (garda moare şi nu se predă). Pentru aceasta trebuie să ne păstrăm cititorii şi n-o putem face decît într-un singur fel: oferindu-le ceva de care ei au nevoie. Acest ceva este informaţia, la zi, de calitate, interesantă, cuprinzătoare. Încercăm şi în acest număr acest lucru. Revenim cu o aducere la zi în ceea ce priveşte sistemele de operare şi adaptoarele grafice. Aruncăm o privire şi în curtea vecinului şi vă prezentăm familia de calculatoare Macintosh în intenţia de a vedea ce se întîmplă şi în lumea calculatoarelor personale care nu sînt compatibile IBM. Facem de asemenea o incursiune în lumea harddisk-urilor şi vă prezentăm pe scurt limbajul de programare C.

Nu dorim să destabilizăm, aşa că dacă nu aţi încheiat încă negocierea salariilor vă rugăm să nu citiţi articolul "60.000 DM deja pentru începători".

Cît de mult am reuşit să ne apropiem de scopul propus rămîne la aprecierea D-voastră. Noi vă semnalăm doar instaurarea Nimicului. "La început era doar ceva foarte mic, o nimica toată, cît un ou de bibilică. Dar astfel de locuri s-au tot lăţit, şi dacă din greşeală intra cineva cu piciorul acolo, dispărea piciorul - sau mîna - sau orice intra acolo. De altfel nu doare, numai că respectivului îi lipseşte brusc o bucată din el."

ing. Romulus Maier

(*) Michael Ende - "Povestea fără sfîrşit"

Cuprins

Ştiri

- Adio ASCII pag.4
- Netware unificat pag.4
- Noul Netware dintr-o privire pag.4
- ş ţ ă î pag.5
- ROMORGADATA pag.6
- Perestroika pag.6
- În curînd: Notepad + Pen pag.6

Magazin

Salarii: 60.000 DM pentru începători . pag.7

Mouse-ul: scurt istoric pag.8

Păcatele scumpe ale conducătorilor oficiilor de calcul pag.52

Hardware

De la 80386 la 80486 . . . . . pag.10

Vă prezentăm:
familia Apple Macintosh... . . . . pag.11

Joystick pag.16

Cu calculatorul te poţi şi juca, dar pentru aceasta pentru anumite jocuri ai nevoie de un joystick. În cele ce urmează o să încercăm să vă explicăm ce este şi cum funcţionează un joystick.

Evaluarea performanţelor
harddisk-urilor pag.22

Standarde grafice

8514/A versus Tiga pag.18

Concurenţa în domeniul adaptoarelor (cartelelor) grafice este puternică. XGA se află în faţa uşii. Favoritul branşei a fost pînă de curînd Tiga. Un standard de facto cum este 8514/A mai are vreo şansă ? În ce direcţie se îndreaptă dezvoltările ?

Merită să mizezi pe XGA ? pag.20

Software

DR-DOS 5.0 vs. MS-DOS 5.0 . . . . . pag.29

OOP - Modă sau ... ? pag.31

"C" - Cuceritorul pag.33

Ce anume determină atracţia crescîndă exercitată de acest limbaj de programare? De ce tocmai "C" ?

Dosar: Turbo C++ 1.0 . . . . . pag.35

Paradigma Paradox pag.38

Înainte să programaţi în PAL (Paradox Application Language), trebuie să înţelegeţi modelul Paradox.

Cursuri

OOP - Partea a treia pag.41

Tema principală a acestei a treia părţi a cursului o constituie metodele virtuale. Cu ajutorul lor, în ultima parte a acestui curs, vor fi tratate structurile dinamice.

Reţele neuronale - partea a treia . . . pag.46

Sistemele expert acumulează fapte şi reguli. în bazele lor de cunoştinţe; pornind de la acestea şi folosind strategii, ele raţionează.

Practică

- Parolă variabilă pag.49
- Protecţia programelor pag.50
- Reset la imprimantă pag.50
- Ieşire în DOS pag.50
- Există o dischetă în unitate ? pag.51
- Autoverificare contra viruşilor pag.51
- Pointeri FAR şi HUGE în C pag.51

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Editorial . . . . . pag. 1
- Jargon . . . . . pag. 53
- Din scrisorile primite la redacţie pag.54
- Mica publicitate pag.54

Adio ASCII

Producători de vîrf din domeniul calculatoarelor vor să înlocuiască în scurt timp setul de caractere ASCII; cu "Unicode", setul ISO ar fi astfel dintr-o dată învechit.

Mai ales la legături care trec de hotarele unei ţări, setul de caractere ASCII impune restricţii partenerilor de comunicaţie. În prezent, cîţiva dintre cei mai importanţi producători de calculatoare vor să prezinte o alternativă: "global computer code for storage and transmission of text around the world". Întreprinderile au fondat Unicode Inc., care va promova setul de caractere universal cu acelaşi nume. Codul de 16 biţi folosit de Unicode asociază fiecărui semn utilizat în prezent un număr şi permite codificarea a circa 65.000 de semne, care ar acoperi "toate limbile vii mai importante" folosite în lume. Inclusiv unele idiomuri din Japonia, China, Taiwan şi Coreea, semnele scrise din alfabetele chirilic, ebraic, arab, grec şi sanscrit. În afară de acestea, noul set conţine simboluri matematice şi tehnice, precum şi semne diacritice pentru deasupra sau dedesubtul literelor, precum şi coduri de comandă (ca de exemplu "avans linie"). Dacă Unicode se impune, munca de ani şi ani de zile depusă de ISO (International Standard Organization) va deveni inutilă. Firme importante (Xerox, Apple, IBM, Microsoft, Sun, Novell, Aldus, Go şi Next) şi-au declarat sprijinul pentru Unicode.

Netware unificat

La CeBit '91, Hanovra, Novell a anunţat versiunea v2.2 a sistemului său de operare în reţea, Netware. Un interes deosebit prezintă faptul că prin aceasta, produsele de pînă acum (ELS I, ELS II, SFT şi Advanced Netware) sînt reunite pentru prima dată într-o unică serie de produse.

În viitor, un potenţial cumpărător nu va mai trebui să-şi elaboreze cu minuţiozitate decizia de achiziţionare a unei reţele de 4, 8 sau nr. maxim posibil de posturi de lucru. Ceea ce pînă acum nu era un banal amănunt tehnic, ci o decizie complicată şi foarte serioasă.

Căci versiunile pentru 286 de pînă acum difereau nu numai după numărul de utilizatori, ci şi calitativ; erau scrise fiecare în alt cod şi nu permiteau treceri de la o variantă la alta decît prin înlocuire.

Trecerea la Netware v2.2, care va putea fi achiziţionat în versiuni pentru 5, 10, 50 şi 100 de utilizatori, va uşura nu numai viaţa utilizatorilor, ci şi a producătorilor de reţele. Căci organizarea service-ului şi întreţinerii pentru un singur produs sînt mai ieftine şi mai sigure pentru un singur produs decît pentru patru.

Astfel că despărţirea de versiunile de Netware de pînă acum, care nu se mai "produc" din ianuarie, va fi fără regrete. Că acest lucru nu va fi adevărat şi pentru vînzători şi revînzători, pare evident. Căci instalarea mai complicată, extensia, service-ul, etc., însemnau desigur afaceri mai mari. Toate acestea din aprilie/mai anul acesta, se vor fi terminat, odată cu lansarea noii versiuni.

Prin restructurarea versiunilor pentru 286, Novell modifică semnificativ tocmai acea serie de produse din care pînă acum s-au făcut cele mai multe vînzări. Căci, măsurînd în bucăţi, ELS I, ELS II, SFT şi Advanced Netware reprezentau circa 90% din vînzări (celelalte 10%: Netware 386).

Văzută dinspre piaţă, introducerea versiunii noi, unificatoare de Netware trebuia să se întîmple chiar de mai mult timp. Căci concurenţa (fie LAN-Manager, fie Vines) oferă deja de timp îndelungat o versiune funcţional echivalentă şi diferind numai prin numărul de utilizatori.

Noul Netware dintr-o privire

Odată cu unificarea liniei sale de produse pentru 286, Novell a introdus cîteva îmbunătăţiri. Cele mai importante sînt:

- instalare simplificată. Procesul de instalare de pînă acum (Netgen) a fost înlocuit printr-un program de instalare complet nou. >Install< cere utilizatorului cîţiva parametri şi face apoi singur configurarea. Procesul de instalare (numit de Novell "intuitiv") e sprijinit de help-uri detaliate, apelabile oricînd cu F1.

- manuale mult îmbunătăţite. Analog instalării mult simplificate, şi manualele s-au subţiat mult. Iar un "Getting started" de 26 pagini va permite o orientare rapidă a utilizatorului.

- Help on-line, prin care utilizatorul are acces direct la documentaţie, ceea ce în marea majoritate a cazurilor ar trebui să facă superflue căutările în manuale.

- structură îmbunătăţită a memoriei file-serverului, ceea ce face posibile mai multe FSP-uri (File Service Process) simultan.

- suport extins pentru Macintosh. Noua versiune Netware poate conlucra cu Appletalk Phase 2, Appletalk Filing Protocol v2.0 şi Tolkentalk.

- opţiuni îmbunătăţite pentru imprimantă. Serverul de imprimantă este conţinut în Netware 2.2, ca VAP şi Exe. S-au făcut îmbunătăţiri, s-au introdus comenzi noi, etc.

- suport extins pentru staţii de lucru. În această categorie intră posibilitatea de a face automat un "Workstation Update"; funcţii noi la suportul de workstation pentru DOS, Macintosh şi OS/2.

- Suplimentar în Netware v 2.2 sînt integrate drivere de 3COM, s-au făcut modificări în "trustee rights" şi au fost operate îmbunătăţiri în utilitare ca FConsole, Login, Map şi Filer.

HiTec
Electronic Engineering SRL

ş ţ ă î
sau
Computerele şi Limba Română

Întrebarea dacă şi la noi calculatoarele vor ajunge vreodată să joace vreun rol în viaţa şi activitatea noastră nu mai are sens. Este clar că avem nevoie de ele. Ce ne facem însă că nu prea ştiu să vorbească româneşte, iar cînd o fac, vorbesc atît de peltic, încît nici nu ştim să rîdem sau să plîngem. Evident, pînă acum s-a recurs la multe soluţii, care mai de care mai ingenioasă, mai simplă sau mai complicată, şi mai ales mai "incompatibilă". Fără a avea pretenţia de a fi găsit cea mai bună soluţie, în urma studiilor noastre s-au conturat cîteva sugestii-rezolvări.

Pentru ca un calculator să poată fi folosit şi în limba română, este necesar înainte de toate să codificăm cumva şi literele româneşti în setul de caractere cunoscut de calculator. Este esenţială stabilirea de comun acord şi acceptarea la nivel naţional a unui asemeni set românesc de caractere, pentru a asigura compatibilitatea între utilizatori. Pe MS-DOS există deja un mecanism de comutare a seturilor de caractere (code page system), dar literele româneşti nu au fost prevăzute în nici unul din seturile disponibile. Două sînt seturile de interes pentru noi: setul obişnuit (437) şi setul internaţional extins (850, care sacrifică o parte din caracterele semigrafice în favoarea unor litere speciale). Deoarece introducerea literelor româneşti în tabelă nu se poate face decît prin renunţarea la alte caractere, este foarte importantă alegerea judicioasă a acestora. Propunerea noastră este următoarea:

(0x83; 131) â
(0x86; 134) ă
(0x8C; 140) î
(0x8F; 143) Ă
(0x91; 145) ţ
(0x92; 146) Ţ
(0xA6; 166) ş
(0xA7; 167) Ş
(0xA8; 168) Â
(0xAD; 173) Î

Această amplasare conduce doar la pierderi minime de caractere (cîteva litere din alfabetul ţărilor nordice şi două semne de punctuaţie din limba spaniolă). În plus această substituire este posibilă în ambele tabele, atît cea normală cît şi cea internaţională, conducînd la apariţia a două seturi de caractere: românesc şi românesc extins.

A doua mare problemă este claviatura. Fiecare ţară europeană a reinventat, cu mai mult sau mai puţin succes, o altă aşezare a tastelor (vezi claviatura germană foarte dificil de utilizat pentru orice altceva decît editare de texte). Dificultatea proiectării unei claviaturi româneşti provine şi din faptul că două dintre cerinţe sînt contradictorii: una impune ca literele româneşti să fie amplasate în dreapta, în continuarea celorlalte litere (ca la maşina de scris: <K L Ş Ţ> <O P Ă Î>) iar cealaltă cere ca semnele de punctuaţie, parantezele drepte şi acoladele să rămînă în aceeaşi poziţie. Rezultă clar că o claviatură românească tebuie să aibă două moduri comutabile funcţie de activitatea preponderentă: dezvoltare de programe sau editare de texte. În mod normal, claviatura va genera aceleaşi coduri ca o claviatură americană, cu menţiunea că folosite împreună cu Alt sau Alt+Shift, tastele din dreapta vor genera codurile literelor româneşti. În modul editare de texte, funcţia acestor cinci taste se inversează, generînd direct <ş ţ ă î>, semnele de punctuaţie fiind generate doar apăsînd simultan Alt sau Alt+Shift.

Afişarea în ecran, pentru modul alfanumeric obişnuit, nu este o problemă, plăcile EGA şi VGA permiţînd definirea completă a setului de caractere.

O ultimă problemă importantă este afişarea pe imprimantă. Pentru imprimantele matriciale compatibile IBM cu set de caractere definibile prin soft, nu sînt dificultăţi majore, putîndu-se încărca în imprimantă imaginile celor zece caractere speciale româneşti. De asemenea, în cazul imprimantelor cu laser compatibile Adobe PostScript, se pot redefini destul de uşor fonturile de caractere, încît literele româneşti să apară corecte la orice dimensiune, înclinare, direcţie, ş.a.m.d., cu inegalabila calitate proprie fonturilor Adobe.

După cum am spus şi la început, aceasta nu se vrea soluţia definitivă a problemei literelor româneşti pe calculatoare. Mai rămîn destule pete albe în dosarul acestei probleme (ca de exemplu modificarea nenumăratelor fonturi folosite de programele care lucrează în mod grafic, cum ar fi Windows 3.0, Ventura Publisher, etc., sau a imprimantelor cu laser care nu folosesc limbajul PostScript). Ceea ce ne dorim să realizăm este o incitare la discuţii, cu convingerea că în cele din urmă hotărîrea ce se va adopta se va respecta la nivel naţional şi va satisface cele mai exigente preferinţe.

În prezent firma HiTec are în curs de dezvoltare extensia cu literele româneşti pentru editorul WordStar 6.0, propunîndu-şi să realizeze şi o versiune pentru Microsoft Word.

Pentru orice lămuriri, sugestii şi propuneri, ne puteţi contacta la următoarea adresă:

HiTec Electronic Engineering SRL R-70317 Bucureşti, Str. Gen. Ipătescu Nr. 4 Tel. (90) 426100 Tlx. 11391 global r

ROMORGADATA

Desfăşurată între 7.-10.05.1991, în Complexul Expoziţional din Piaţa Scînteii (pardon, Presei Libere), expoziţia specializată organizată de ROMEXPO S.A. şi I.E.G. Salingen a fost, se pare, un succes. Cel puţin pe partea care ne interesează - PC-uri & Co.

Au fost prezentate multe exponate, majoritatea de vîrf, şi s-au putut face contracte pe loc. De la imprimante laser HP Laser Jet IIID şi calculatoare de elită ca HP Vectra 486/33T pînă la banale (dar importante) casete de toner, dischete sau hîrtie xerox, cartele de extensie de tip fax/modem, soundblaster sau laptop-uri Toshiba 486 - cam toată panoplia PC-urilor a fost prezentă. Nu în ultimul rînd Borland a stîrnit interes, atît prin "programul de amnistie" anunţat cît şi prin preţurile în general accesibile pentru marea majoritate a produselor. Şi au putut fi văzute - şi cumpărate - ultimele noutăţi; inclusiv Borland C++ 2.0 (143.055 lei la Logic) sau Novell Netware v2.2 (188.605 lei pentru 5 utilizatori). Se pare că a venit vremea cînd - pe lei! - se poate cumpăra (aproape) orice hard sau soft. Timpuri noi?

Perestroika

Joc de acţiune pentru EGA/VGA - primul joc shareware din URSS! Un joc pasionant, cu multă acţiune, efecte sonore comice şi o grafică foarte bună. Jocul are 25 de nivele de joc şi necesită din partea dumneavoastră capacitate de reacţie rapidă şi nervi tari. Scopul jocului este de a depăşi cu micuţa broscuţă verde a perestroicii (GORBI) multe obstacole mici, mobile şi de a aduna puncte. Cu cît creşte nivelul de joc, creşte şi gradul de dificultate şi dintr-o dată apar broaşte uriaşe, urîte şi roşii (STALINOS) care vă vor viaţa. Perestroika, împreună cu Commander Keen şi Shooting Gallery, sînt considerate cele mai bune jocuri shareware ale anului 1990. Pe cînd la noi...? Jocul poate fi comandat sub codul S666 de la Software IHEK Rubröder, Ermener Weg 3, D-W4503 Dissen, Fax + Tel. 05421 - 2231, ptr. circa 12 DM.

Creionul electronic ca dispozitiv de intrare.

notepad-urile nu vor avea nevoie de tastatură sau mouse.

În curînd: Notepad + Pen

După laptop şi notebook, un nou tip de calculatoare se pregătesc să intre în competiţie: notepad-urile. Cu greutăţi maximale de 2 kg, avînd dimensiunea ideală de mărimea unei foi A4, cu o înălţime care în cele mai avantajoase cazuri nu va depăşi 2 cm. Aceste dimensiuni devin posibile prin harddisk-uri din ce în ce mai mici; cele mai noi produse Quantum de ex. au o înălţime de numai 1cm, la o capacitate de 40 Mbyte.

Revoluţionară este (va fi) interfaţa om-calculator: "paperlike interface". Ca pe o hîrtie, utilizatorul scrie cu creionul electronic (stylus) pe o tabletă digitizoare transparentă, montată deasupra displayului propriu-zis, realizat cu cristale lichide. Tableta are un strat care permite determinarea exactă a poziţiei actuale a creionului în momentul cînd se apasă cu acesta asupra tabletei. Drumul parcurs de creion pe tabletă este afişat - asemănător scrisului normal - pe display. IBM apreciază că într-o zi acest gen de comunicare om-calculator va fi la fel de important cum este în prezent sistemul cu tastatură şi mouse.

Microsoft a făcut recent cunoscută o listă de producători de calculatoare pentru a semnaliza proiectanţilor de soft că pentru "Pen Windows" există o largă bază hardware. Un total de 21 de producători de calculatoare sau echipamente periferice şi-au făcut cunoscut sprijinul pentru noua suprafaţă - sistem de operare a Microsoft-ului. Lista conţine aproape toate "numele" din extremul orient, printre ele Toshiba, NEC, Sharp, Mitsubishi, Sanyo, Seiko, Epson, Kyocera, Fujitsu şi Samsung. Canon, Hitachi şi OKI plănuiesc dezvoltarea de 'creioane" pentru Pen Windows. NCR, Wang, Grid precum şi producătorii de periferice Calcomp şi Summagraphics şi-au declarat deasemenea sprijinul. Lipsesc de pe lista Microsoft giganţi ca IBM, Compaq şi Hewlett-Packard.

60.000 DM deja pentru începători

Salariile din industria calculatoarelor cresc vertiginos. Absolvenţii unei şcoli de profil n-ar trebui să semneze contracte de muncă cu un salariu anual mai mic de 60000 DM. Salariul mediu pentru un începător se afla la începutul anului 1991 la 63800 DM, conform unei anchete efectuate de întreprinderea de consultanţă Interconsult din Germania.

În total 116 întreprinderi din industria electronică sînt chestionate anual de Interconsult, din Vaihinger, în legătură cu mărimea salariilor pe care le plătesc. La ancheta din ianuarie şi februarie 1991 au participat 116 întreprinderi din industria electronică, dintre care 96 erau producători iar 20 distribuitori.

În urma chestionarului a rezultat că salariile din industria calculatoarelor vor creşte şi în continuare în favoarea angajaţilor. Conducătorii afacerilor caselor de hardware şi software PC au obţinut, conform aşteptărilor, salarii de vis. Salariile lor au crescut faţă de anul anterior cu 8000 DM ajungînd la 262000 DM anual.

Cîştiguri mai mari pentru superiori

Ancheta celor din Vaihing s-a terminat cu rezultate îmbucurătoare şi pentru absolvenţii de licee şi universităţi. Chiar şi începătorii în meserie au cîştigat la începutul lui 1991, în medie, mai mult de 60000 DM. Chiar şi în anul precedent această limită a fost depăşită şi s-a ajuns la 60500 DM salariu anual, iar în 1991 media salariului anual va ajunge la 63800 DM.

Şansa unui începător de a cîştiga în anii următori mai mult decît 60000 DM anual depinde în mare măsură de capacitatea lui de a ajunge după un timp superiorul unor colegi din echipă. Acest lucru se poate observa la prima vedere din tabelele furnizate de Interconsult.

Cei care lucrează de ex. în ramura sistemelor cu microprocesoare, ca ingineri proiectanţi, se pot aştepta la un salariu anual între 60000 DM şi 65000 DM, dar acumulînd experienţă în domeniu, în cîţiva ani, pot să obţină cu 20000 - 30000 DM mai mult. Dacă în decursul anilor începătorul devine şef de secţie, şi va avea deci o funcţie de răspundere, atunci saltul va putea fi mai mare.

Un inginer proiectant care concomitent este şi şeful unei secţii cu cel puţin 6 ingineri, va putea cîştiga cel puţin dublul salariului său de începător.

În general în industria electronică s-au încetăţenit o serie de prestaţii secundare care se plătesc în afara salariului tarifar negociat de sindicate. Astfel 88% dintre producători şi 71% dintre distribuitori plătesc pentru angajaţii lor fonduri de pensii care le va asigura o pensie suplimentară pe lîngă pensia asigurată legal.

Asigurări sociale

Şi în cazuri de îmbolnăviri angajaţii din industria electronică sînt asiguraţi mai bine decît prevede legea. 88% dintre întreprinderi garantează plătirea în continuare a salariului peste termenul limită de 6 săptămîni prevăzut de lege, unele chiar pînă la 2 ani.

Cca. 96% dintre cei care se ocupă de relaţiile întreprinderii cu exteriorul sînt cuprinşi în această reglementare; în plus aceştia sînt asiguraţi pentru caz de deces sau de invaliditate, cel mai adesea cu poliţe de asigurări pe viaţă, care se ridică la 100000 DM şi care, de regulă, sînt valabile 24 ore/zi şi nu doar pe timpul serviciului.

Concediul anual variază între 29 şi 32 de zile, la fel ca în cele mai multe ramuri ale industriei.

(I. M.)

Salarii

Sisteme dotate cu microprocesor

Servicii interne (post de lucru fix)	mii DM/an
Inginer proiectant, absolvent	60-65
Inginer proiectant, 2-4 ani vechime	71-83
Inginer proiectant, vechime > 4 ani	85-94
Şef echipă proiectare cu 2 sau mai mulţi tehnicieni	95-119
Şef proiectare cu mai mult de 6 ingineri	123-146
Inginer de aplicaţii, pînă la 3 ani vechime	77-93
Inginer de aplicaţii, mai mult de 3 ani vechime	98-120
Inginer producţie-marketing, pînă la 3 ani vechime	87-103
Inginer producţie-marketing, mai mult de 3 ani vechime	111-129
Director producţie-marketing (director tehnic)	125-143
Servicii externe (muncă de teren)	
Inginer expert într-un domeniu, pînă la 3 ani experienţă în producţie	76-94
Inginer expert într-un domeniu, mai mult de 3 ani experienţă în producţie	100-118
Inginer comercial, pînă la 3 ani în serviciul extern	102-111
Inginer comercial, mai mult de 3 ani în serviciul extern	115-129
Şef comercial zonal, şef al unui inginer	137-160
Şef comercial zonal, şef a cel puţin 3 ingineri	167-224
Director	238-304

Oameni şi şoareci

Recent, mouse-ul, fără de care lucrul cu un PC a devenit de neconceput, şi-a sărbătorit a 28-a aniversare. Motiv suficient pentru a scotoci puţin prin istoria lui.

Şoarecii timpurii

"În vremea aceea, în urmă cu 28 de ani, experimentam mai multe aparate de introducere date. De fiecare dată mouse-ul s-a dovedit a fi foarte folositor, astfel încît comparîndu-l cu alte instrumente de introducere, a cîştigat întrecerea". ne spune Doug Engelbart, inventatorul mouse-ului.

Prototipul din lemn al primului mouse, pe care l-a creat Doug Engelbart la Institutul de Cercetări Stanford, în 1963, a fost conceput pentru calculatorul Augment. Ulterior diferite generaţii de PC-uri s-au lăsat influenţate de ideile lui Engelbart, începînd cu Star-ul firmei Xerox, prin Lisa lui Apple şi pînă la Macintosh-urile firmei Apple. Dar Engelbart nici nu şi-ar fi putut închipui ce avînt va lua invenţia sa.

Mouse-ul inventat de Engelbart era un simplu instrument de introducere, analogic, care transmitea soft-ului cîte un semnal la fiecare mişcare a mouse-ului, influenţînd mişcarea cursorului. În interiorul acestui corp de lemn se aflau două rotiţe metalice, legate între ele prin două rezistenţe variabile.

Dar ideea de a lucra cu mouse-ul a devenit cunoscută abia în momentul în care, la începutul anilor 70, cunoscutul centru de cercetări Palo Alto Research Center (PARC) al firmei Xerox l-a însărcinat pe Jack S. Hawley să construiască prima variantă digitală a "şoricelului" lui Engelbart.

În paralel Xerox-ul a dezvoltat calculatorul Alto care urma să fie livrat împreună cu mouse-ul respectiv. Cu toate că din acest tip de calculator au fost vîndute mai puţin de 100 de exemplare, drumul pentru dezvoltarea calculatoarelor personale şi al mouse-urilor a fost deschis.

La cererea firmei Xerox, Hawley elaborează în 1975 un nou standard de mouse, care a fost adoptat de mulţi producători şi care s-a menţinut mult după anii '80. În continuare Hawley lucrează la propria sa firmă "Mouse House" din Berkeley (California), la configurarea şi producerea de mouse-uri.

După ce mouse-ul firmei Xerox a trezit tot mai multă atenţie, Microsoft a hotărît să creeze un mouse propriu. Între timp la Microsoft a ajuns un fost colaborator al firmei Xerox, pentru a permite operarea la un nou produs, Microsoft Word, cu ajutorul mouse-ului. În acelaşi timp Bill Gates, Paul Allen şi Raleigh Roark discutau extinderea paletei de produse Microsoft cu un produs hardware adecvat. Astfel s-a pus baza primului mouse Microsoft.

Dintr-un boţ de lut

Cu concepţia primului mouse Microsoft a fost însărcinat un designer, care a construit conform dorinţei colectivului Microsoft, un model din lut care era prevăzut în partea de dedesubt cu pioneze care înlocuiau partea glisantă.

Strămoşul din lut/lemn

Raleigh Roark îşi aduce aminte de prezentarea acestui mouse: "Unii dintre noi stăteau ceasuri în şir la masa de conferinţe, ocupaţi cu mişcarea înainte şi înapoi a acelei grămăjoare de lut, pentru a afla dacă ne place sentimentul pe care ni-l dă această mişcare. Dar nici unul dintre noi nu era pe deplin mulţumit, astfel încît i-am schimbat înfăţişarea şi dimensiunile după imaginaţia noastră. Cu un nou model de lut în mînă am zburat apoi la Tokio pentru a da "chestia" unui producător de electronică s-o construiască".

Roark a fost însoţit de Kay Nishi, care pe atunci era vicepreşedintele Microsoft şi preşedinte al comisiei ASCII din Japonia. Nishi şi Roark s-au întîlnit acolo cu producători de aparatură electronică pentru a discuta ideile lor. Discuţiile s-au sfîrşit

repede şi brusc, producătorii n-au crezut că ideea celor de la Microsoft ar fi realizabilă. Ei au avut rezerve că unitatea de codificare ar putea încăpea în acea carcasă minusculă.

Roark povesteşte: "s-a discutat mult, de ce acest lucru ar fi imposibil, important era doar că nu se putea. Pînă cînd la un moment dat s-a făcut o linişte deplină şi inginerul şef ne-a comunicat că se va retrage timp de o oră, după care ne va da o soluţie. Într-adevăr inginerii au revenit cu o propunere de carcasă care avea capacitatea de a funcţiona, iar cîteva luni mai tîrziu Microsoft poseda primul mouse".

Prima generaţie

"Crearea mouse-ului serial era pentru mine o problemă captivantă, o deschidere a concepţiilor, nerealizată de nimeni pînă atunci", spune Raleigh Roark conducătorul secţiei de dezvoltare pentru un mouse serial a Microsoft-ului.

În 1983 Microsoft prezintă un nou produs pentru IBM-PC, denumit Microsoft busmouse, un mouse mecanic avînd două taste, sprijinindu-se pe o sferă de oţel şi cîteva role, care înregistrau mişcările mouse-ului pe orice suprafaţă. "Îndemînarea" lui era dictată de o cartelă de 8 biţi, care era echipată cu un cip Intel 8255, ca interfaţă periferică programabilă, şi cu cîteva cipuri auxiliare. Acest tip de mouse avea avantajul că nu consuma multă energie electrică şi deci nu avea nevoie de o sursă distinctă de alimentare.

La un an de la apariţia busmouse-ului Microsoft a conceput o variantă serială de mouse, atingînd un stadiu tehnologic avansat, deoarece putea fi cuplat direct la interfaţa serială RS232. Acesta nu avea nevoie nici de o placă de bus nici de alimentare separată, deoarece prin intermediul unui procesor CMOS integrat putea aborbi energia necesară prin interfaţa RS232.

Odată cu apariţia, în anul 1983, a sistemului de operare MS-DOS 2.0, mouse-ul a avut un cuvînt important de spus. MS-DOS era echipat cu un aşa-zis driver de periferic instalabil la dorinţă, ceea ce simplifica foarte mult configuraţia oricărui calculator echipat cu sistemul de operare DOS şi cu un mouse.

În 1985, la apariţia AT-urilor IBM, odată cu creşterea numărului de cartele grafice de înaltă rezoluţie s-a ţinut cont şi de cele două noi versiuni de drivere.

Generaţia a doua

Mouse-ul firmei Microsoft cu o rezoluţie de 200 puncte per inch a schimbat modul de utilizare al primului mouse. Dublarea rezoluţiei a avut ca efect faptul că utilizatorul nu mai trebuia să plimbe mouse-ul pe întreaga masă de scris pentru a mişca cursorul pe ecran.

Steve Shaiman proiectantul principal al softului pentru mouse-ul 5.0 al firmei Microsoft ne spune: "În 1985 a fost prezentată o altă variantă a mouse-ului Microsoft. Modificările erau observabile la prima vedere: culoarea gri a tastelor care atingeau acum marginile, carcasa nou conceputâ şi bila îmbrăcată în cauciuc în locul celei îmbrăcate în oţel. Adevărata diferenţă însă s-a simţit, nu s-a văzut. Cu dublarea rezoluţiei la 200 dpi (dot per inch - puncte pe inch), mouse-ul putea fi manevrat mult mai uşor, necesita o rază de acţiune mult mai mică pentru deplasări (aprox. 10-12 cm); cele mai multe probleme puteau fi rezolvate rapid şi cu economie de mişcare. În comparaţie cu acesta modelul anterior părea greoi şi grosolan, mai ales datorită razei mai mari de acţiune (aprox. 20-25 cm).

În luna mai 1986 Microsoft a ieşit pe piaţă cu un mouse care era o versiune modificată a interfeţei busmouse, bazat pe aşa-numitul cip "Inport". Deoarece noul driver de mouse folosea sistemul de întreruperi programabil al acestui cip, viteza de reacţie a mouse-ului a putut fi ridicată vizibil.

Generaţia a treia

"Noul mouse Microsoft (care arată ca o bucată de săpun), cu noua sa bilă poziţionabilă şi cu mecanica îmbunătăţită este cu adevărat diferit de predecesorul său". afirma John C. Dvorak, în PC Magazine, 22.12.1987.

Mouse-ul celei de-a treia generaţii a fost prezentat de Microsoft în 1987. Corpul devenise mai suplu, mai îngust iar clicul tastelor mai plăcut. Arhitectura internă a acestui mouse tot se mai orienta încă după cea a mouse-ului cu taste gri. Cîteva modificări esenţiale au dus la o controlabilitate mai uşoară; de ex., poziţionarea bilei (care înainte se afla în spate) în faţa mouse-ului şi mărirea tastei din stînga faţă de cea din dreapta.

Generaţia a patra

Actualul mouse, prezentat în 1989, de culoare albă şi cu o rezoluţie de 400 dpi aparţine generaţiei a patra. Aspectul este similar cu cel din generaţia a treia, dar tehnologic se află pe un plan net superior. Importante sînt următoarele aspecte: rezoluţia net îmbunătăţită pe de o parte, iar pe de altă parte sensibilitatea şi viteza reglabilă. În plus în carcasa mouse-ului a fost introdusă electronica de control pentru mouse-ul serial PS/2, pentru a optimiza manipularea produsului.

"Odată cu apariţia soft-urilor tot mai complexe va trebui să avem tot mai multe aparate de introducere eficiente în lucrul cu calculatorul. În viitor va exista în mod cert un mouse". - spune Cary Lu, autorul cărţii "The Apple Macintosh Book", ediţia a treia, Microsoft Press.

Astăzi deja un soft cu suprafaţă utilizator grafică este mai degrabă o regulă decît o excepţie, şi în paralel cu această dezvoltare acceptarea mouse-ului se extinde continuu:
Bun venit în era şoarecilor !

(I. M.)

De la 80386 la 80486

PC-urile cu 80486 nu sînt chiar cele mai ieftine. Acest articol încearcă să explice de ce costă mai mult decît PC-urile cu 80386 şi care sînt avantajele noului procesor.

Pînă la apariţia lui 80486 fiecare nou procesor Intel (8086, 80286, 80386) era o veritabilă construcţie nouă. Comenzile fiecăruia menţineau compatibilitatea cu procesoarele anterioare, deci programele scrise pentru tipuri de procesoare mai vechi sînt funcţionale şi pe cele noi. Cu totul altfel se prezintă situaţia la procesorul 80486 care nu este o construcţie cu totul nouă: într-un singur cip el conţine toate componentele pe care le conţinea predecesorul său, procesorul 80386, coprocesorul 80387 şi controller-ul cache 82385.

80386 posedă, în comparaţie cu predecesorii săi, avantaje importante care în ciuda "folosirii sale îndelungate" (din 1985 şi pînă astăzi) nu au dus la necesitatea unei construcţii noi. El poate fi folosit la o frecvenţă de tact de 33 MHz şi se distinge prin viteza sa mare de calcul. Tactul maximal permis pe magistrală este de 66 MHz.

Lăţimea structurii componentelor realizate prin aşa-numitul procedeu CHMOS-III este de numai 1.5 μm, ceea ce înseamnă că fiecare conductor şi spaţiile dintre conductori au o lăţime de numai 0.0015 mm. Din acest motiv pentru producerea lor sînt necesare aparate foarte precise. Cipul conţine 275000 tranzistori. Carcasa, aşa-numită Pin-Grid, conţine 132 de pini şi are dimensiunile de 36 x 36 mm.

Procesorul i486 conţine aproape de patru ori mai multe tranzistoare (1.2 milioane) şi din acest motiv este fabricat prin procedeul 1-μm-Standard-CHMOS-IV (conductorii şi spaţiile dintre ei au o lăţime de doar 0.0001 mm). Carcasa din ceramică are dimensiunile de 44.8 x 44.8 mm şi posedă 168 pini de legătură în partea inferioară.

În concepţia procesorului 486 au fost preluate structurile de bază ale fiecărui procesor în parte (80386, 80387 şi 82385), dar ele au fost şi modificate pe alocuri. Astfel partea de procesor şi coprocesor au fost legate printr-o magistrală dublă de 32 de biţi, dispărînd astfel "locul îngust" existent pînă acum între cele două procesoare. Aşa se explică şi faptul datorită căruia un 80486 lucrează mai rapid decît un 80386 în tandem cu un 80387.

O particularitate a lui 80386 a fost păstrată şi la 80486. Unitatea internă de management a memoriei (MMU - Memory Management Unit), uşurează sub MS-DOS, dar mai ales sub Unix, lucrul cu pachete de programe mari. Ea controlează atît spaţiul de memorie fizic cît şi pe cel virtual.

Mărimea memoriei fizice atît a procesorului 80386 cît şi a lui 80486 este dată de cei 32 de conductori de adresare şi este de 4 GBytes. Spaţiul logic (virtual) de adrese este de 64 TBytes (1 Terabyte = 1000 Gigabytes). Nici 4 GBytes aproape că nu pot fi folosiţi, darămite 64 TBytes !

Spaţiul de memorie virtual se împarte în mai multe pagini de cîte 4 KBytes. Aceasta este cea mai mică unitate care poate fi transferată. Acest lucru se poate observa foarte bine cu sistemul de operare MS-DOS, un PC 386 şi Windows 3.0. Această suprafaţă utilizator rezervă pe harddisk o aşa-numită zonă de swapare ("Swap Area"), ce corespunde ca mărime cu mărimea memoriei virtuale. Cu cît este mai mare această zonă, cu atît este mai mare şi memoria virtuală care stă la dispoziţia programului.

Pentru a putea utiliza un 80386 la viteza sa maximă de lucru este necesar ca şi periferia să fie tot la fel de rapidă. Din această cauză se încearcă tot felul de trucuri, pentru a intercala cît mai puţine cicluri de aşteptare (waitstates). Unul dintre aceste trucuri este aşa-numitul "Pipelining", în cazul căruia unitatea de management a memoriei foloseşte semitacturile pentru a determina în timp util adresele care urmează. De obicei procesorul află adresa şi definiţia următorului ciclu de bus înainte de prelucrarea ciclului în curs, fapt care îi permite să trimită deja noile adrese pe bus (magistrală). Periferia are astfel timp suficient pentru prelucrare şi va preda datele abia după trei perioade de tact, fără a frîna procesul.

Utilizarea memoriei cache (tampon) la sistemele 80386 place în continuare. Nu este importantă numai mărimea ei ci şi modul de administrare. În această memorie se scriu datele care sînt necesare mai la urmă. Deoarece cel mai adesea un program nu "ţopăie sălbatic" prin tot domeniul de adrese, probabilitatea ca datele din memoria cache să fie necesare de mai multe ori este foarte mare. Această probabilitate, în funcţie şi de modul de administrare, este între 89% şi 97%.

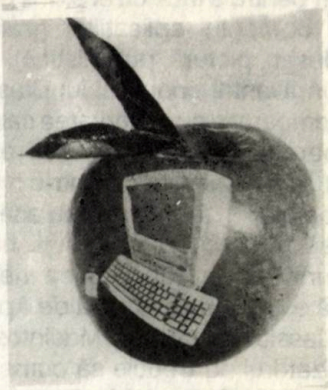
În procesorul 80486 este integrată o memorie cache de 8 KBytes. Administrarea ei se face prin intermediul controller-ului de pe cip, formatul cache avînd chiar şi o lăţime de 128 biţi. Cu toate că cipul 80486 înglobează deja un coprocesor, unele plăci dau posibilitatea instalării unui coprocesor suplimentar Waitek 4167. Programele de calcul intensiv care recunosc acest coprocesor rulează de două, trei ori mai rapid.

Pentru măsurarea vitezei reale a unităţii centrale (CPU - Central Processor Unit) se foloseşte testul Dhrystone, iar pentru măsurarea vitezei cu un coprocesor se foloseşte testul Whetstone. Ambele sînt teste ("Benchmarks") standardizate şi conţin un spectru prestabilit de comenzi de procesare.

O comparaţie între rezultatele testelor pentru un 80486 şi un 80386 (cu 80387) ne arată că primul are performanţe mult mai ridicate, fapt datorat organizării sale optimizate.

(I. M.)

Să-l muşti ...



Macintosh înseamnă o adiere de rafinament şi confort ales (de utilizare a unui calculator) la preţuri medii, acceptabile. Citiţi care Mac vi s-ar potrivi cel mai bine.

Faptul că începătorii în ale calculatoarelor îşi încep cariera lor de utilizatori nu cu puţină teamă, la 10 ani de la introducerea calculatoarelor IBM-PC (1982), nu este fără motiv. Căci ca şi pînă acum cei mai mulţi utilizatori sînt nevoiţi să utilizeze sistemul de operare MS-DOS şi comenzi neplăcute de genul: "XCOPY /A/E/P", "GRAFTABL", sau "EXE2BIN".

De mult există însă o alternativă: deja în 1984 constructorul californian de calculatoare Apple a prezentat un concept de calculator cu un confort utilizator revoluţionar, şi anume Macintosh. Un posesor de Mac nu trebuie să facă risipă nici de acrobaţie mentală şi nici de răbdare pentru a descoperi secretele utilizării calculatorului, ci se poate concentra imediat după pornirea lui asupra muncii propriu-zise şi chiar şi cel mai neştiutor dintre toţi poate lucra după 1/2 oră cu un Macintosh.

Ceea ce face Macintosh-ul atît de atractiv, (Macintosh este şi numele unui sort de mere), este sistemul său grafic de manipulare şi anume "Finder". Indiferent că se doreşte lansarea în execuţie a unui program, ştergerea sau copierea unui fişier, totul se rezolvă printr-un simplu clic de mouse şi prin simboluri uşor de înţeles. Finder permite o utilizare nu numai uşoară ci şi unitară. Toate programele pot fi deservite după aceeaşi schemă. Dacă utilizatorul cunoaşte un program, atunci le cunoaşte pe toate.

Un alt atu: toate calculatoarele Macintosh stăpînesc aşa-numitul multitasking. "Multi" înseamnă "mai multe", "task" înseamnă "sarcină"; calculatoarele care permit multitasking-ul pot prelucra (rezolva) mai multe sarcini în paralel, mai multe programe putînd fiind rulate simultan.

Avantajul: cînd utilizatorul scrie, de ex., o scrisoare unui client şi are nevoie de informaţii despre partenerul de afaceri, poate comuta rapid într-un program bază de date, poate obţine datele despre client şi se poate întoarce în programul de prelucrare de texte - chiar şi schimbul de date între aplicaţii este posibil în acest mod. Cele mai multe PC-uri nu pot face acest lucru nici pînă în ziua de azi. (Odată cu apariţia suprafeţei utilizator MS-Windows 3.0, pe PC-uri lucrurile s-au mai schimbat.)

Pentru începători: Macintosh-Classic

Cine nu s-a născut ca şi copil al unor părinţi bogaţi, mulţi ani nu putea decît să privească cu jind la posesorii de Macintosh-uri: calculatoarele Apple erau pentru majoritatea cumpărătorilor mult prea scumpe.

Totuşi începînd din octombrie, anul trecut, odată cu apariţia noului Macintosh Classic situaţia s-a schimbat. Un Classic fără harddisk (1 MByte memorie principală şi un lector de dischete de 3,5") se poate obţine deja cu 2000 DM, iar unul cu un harddisk de 40 MBytes poate fi cumpărat cu 3000 DM (incluzînd 2 MBytes RAM şi un lector de dischete).

Noul Apple înlocuieşte Mac-urile compacte Macintosh SE şi Macintosh Plus. În exterior li se aseamănă amîndorura, tehnic însă este urmaşul lui SE: Apple construieşte ambele calculatoare constitutiv din aceleaşi componente. Totuşi placa de bază a lui Classic este numai pe jumătate atît de mare cît cea a lui SE - microminiaturizarea electronicii face acest lucru posibil.

Macintosh Classic este acţionat de un procesor Motorola 68000 cu un tact de 8 MHz. Puterea de calcul a lui Classic este suficientă pentru aplicaţii simple, pentru calcule complexe nu ajunge însă (cum ar fi de ex. pentru aplicaţii grafice profesionale sau pentru gestiunea bazelor de date mari).

Classic este o opţiune potrivită pentru începători, pentru lucrul la domiciliu, sau pentru lucrări care nu necesită nici un tempo de calcul intensiv şi nici o prezentare color. Punctele sale forte:

- procesarea de texte
- gestiunea bazelor de date mici (de exemplu cu adrese şi numere de telefon) şi
- calculul tabelar de dimensiuni mici

Numele de un milion de dolari

Atunci cînd şeful firmei Apple John Sculley a vrut să boteze noul Macintosh "Classic" a trebuit să constate că acest nume era protejat deja pentru o altă firmă. Un milion de dolari a trebuit să plătească Apple din această cauză filialei Mercedes "Modular Computer Systems" din însorita Florida, pentru a-şi putea boteza noul produs "Classic". Ironie a sorţii: Classic a fost (şi este în SUA) şi numele unui sort de Coca-Cola. Şeful firmei Apple John Sculley şi-a început cariera managerială la Pepsi-Cola şi şi-a făcut un bun nume în ramură ca expert în marketing.

- De asemenea Classic este indicat pentru a face cu el primii paşi în domeniul aplicaţiilor grafice (desen, pictură, publicistică).

Un avantaj enorm al lui Classic în comparaţie cu majoritatea calculatoarelor compatibile IBM este faptul că poate fi cuplat într-o reţea ("Appletalk"), fără ca pentru aceasta să fie necesare investiţii hard semnificative. Electronica necesară este deja construită de Apple în Classic (ca la fiecare Macintosh). Utilizatorul nu trebuie să cumpere în plus decît un cablu "Localtalk" (preţ: 160 DM) şi software-ul de comandă corespunzător.

Alte posibilităţi de conectare: micuţul Apple are (din nou ca toate Mac-urile) o interfaţă SCSI (Small Computers System Interface - un standard de interfaţare), la care pot fi conectate simultan pînă la şase periferice (de ex. un harddisk suplimentar şi unităţi de salvare/restaurare pe/de pe bandă magnetică).

Dacă lectorul de dischete încorporat (pentru dischete de 3,5" cu o capacitate de 1,44 MBytes) nu este suficient, Classic are o mufă la care mai poate fi conectat un lector de dischete extern.

Macintosh Classic este compact şi are dimensiuni reduse, avantajul: spre deosebire de majoritatea calculatoarelor MS-DOS el poate fi aşezat pe masa de scris şi mai rămîne loc şi pentru acte şi cărţi. Un alt avantaj: deoarece este atît de mic şi de uşor, şi întrucît are integrat în carcasă un mîner, poate fi foarte uşor transportat.

Dezavantaj: deoarece monitorul monocrom este înglobat în carcasă, trebuie să te descurci cu o diagonală a ecranului de doar 9" (23 cm) - pentru comparaţie: diagonala unui monitor normal măsoară cel puţin 12" (30 cm). Consolarea: cu toate că monitorul lui Classic este mic, textele şi imaginile sînt afişate cu rezoluţia ridicată de 342 de linii a cîte 512 puncte fiecare..

O altă lipsă a lui Classic: el nu are nici un slot (mufă de extensie) liber pentru plăci suplimentare cu ajutorul cărora i-ar putea fi extinse capacităţile (de ex. pentru grafică color).

Există un singur slot pentru extensia de memorie numit "systems memory slot". La Classic-urile care au memoria principală de doar 1 MByte, în acesta se poate introduce o cartelă RAM Apple specială care să ridice memoria la 2 MBytes (preţ: cca. 600 DM). Această cartelă mai conţine şi socluri pentru cipuri de memorie, aşa-numite SIMM (Single In line Memory Module) cu ajutorul cărora memoria poate fi extinsă pînă la 4 MBytes. (preţ pe MByte: cca. 150 DM).

Inginerii de la Apple au dotat Classic-ul şi cu o intrare audio la care poate fi conectat un microfon sau o instalaţie stereofonică, pentru a introduce în calculator mesaje verbale, muzică sau sunete, a le prelucra şi a le reda apoi.

Calculatorul: Macintosh SE/30

El este tot atît de mic ca şi modelul Classic, dar în interiorul lui mocneşte puterea unui uriaş: Macintosh SE/30 (preţ: începînd de la 7600 DM) lucrează cu rapidul procesor 68030 (momentan cel mai rapid procesor Motorola aflat pe piaţă) cu un tact de 16 MHz, căreia i se adaugă în cazul calculelor deosebit de complicate şi coprocesorul 68882. Programele de calcul tabelar care recunosc coprocesorul 68882 lucrează cu acesta de cca. 100 de ori mai rapid. El mai este echipat cu o memorie de 2 MBytes, un harddisk de 40 MBytes şi un slot (aşa numit "processor direct slot", mufă cu acces direct la procesor).

Mac SE/30 a fost primul calculator Apple cu Procesor Direct Slot. Plăcile de extensie introduse în acest slot pot accesa direct rapidul procesor 68030.

Modelul SE/30 este recomandat pentru utilizatorii care au nevoie de o putere de calcul ridicată, dar care se pot lipsi de un monitor mare şi de o reprezentare color. Se recomandă pentru:

- procesare de texte
- baze de date mari şi
- programe de calcul tabelar costisitoare, dar şi ca
- server (calculator principal) într-o reţea.

Cine doreşte să realizeze pe SE/30 grafică la nivel profesional

Operare uşoară: Finder

Calculatoarele Macintosh sînt cele mai prietenoase calculatoare din lume în privinţa operării şi acest lucru se datorează suprafeţei grafice Finder (se pronunţă Fainder). După un test al Institului Fraunhofer din Stuttgart, din punct de vedere al operării calculatoarele compatibile IBM nu pot da clasă unui Mac nici chiar cu suprafaţa utilizator Windows.

Inima operării sub Finder: mouse-ul. Dacă, de ex., cursorul este deplasat deasupra simbolului harddisk-ului şi se apasă de două ori tasta mouse-ului se deschide o fereastră cu directorul harddisk-ului.

Chiar dacă nu este la fel de uşor să foloseşti un IBM PC cum este de uşor să foloseşti un Mac, IBM PC prezintă totuşi un avantaj uriaş: pentru ele există cea mai bogată ofertă de software.

Atunci cînd utilizatorii de Mac-uri vor să aibă acces la această ofertă (pentru că au nevoie de ex. de o anumită aplicaţie) apare o problemă: Mac-urile nu pot prelucra programele scrise pentru MS-DOS. Motivul: calculatoarele Apple utilizează un alt sistem de operare (Finder) decît PC-urile compatibile IBM (MS-DOS).

O ieşire din impas există doar pentru posesorii de Mac-uri din puternica serie II (IIsi, IIci şi IIfx) şi de SE/30. Aceştia pot utiliza programul de emulare Soft-PC (900 DM) care emulează un PC/XT şi pot rula astfel toate programele MS-DOS.

va trebui să înlocuiască monitorul de 9" cu un monitor mai mare. Există producători care oferă plăci de extensie cu ajutorul cărora se pot conecta la SE/30 monitoare mari care să poată prezenta una sau chiar două pagini DIN A4.

SE/30 este simpatizat mai ales în SUA şi Japonia. Motivul: în ambele ţări birourile uriaşe sînt divizate în celule de lucru minuscule. Fiecare angajat dispune de un minibirou cu o suprafaţă de cca. doi pe doi metri şi de o micuţă masă de scris. Un PC normal nu prea se potriveşte înăuntru - în asemenea locuri este necesar un SE/30 care ocupă cel mult suprafaţa unei mape de scris DIN A4 şi care dispune totuşi de o putere de calcul maximală.

Curcubeu ieftin: Macintosh LC

Cu Macintosh LC Apple oferă în sfîrşit cumpărătorilor un model color plătibil ("LC" semnifică "Low cost Color", în româneşte "color ieftin", dar ieftin în sensul că merită banii). Preţul unui LC cu monitor de 12": 4900 DM.

Adaptorul grafic care comandă monitorul este înglobat pe placa de bază (motherboard). Cu monitorul Apple color de 12" (rezoluţie 384x512) LC poate reprezenta imagini în 256 de culori, cu un monitor color de 13" (rezoluţie 480x640) poate reprezenta imagini în 16 culori.

Mai multe culori pot fi obţinute doar prin adăugarea unei plăci de extensie suplimentare, o aşa-numită cartelă "Video-RAM", care conţine mai multă memorie pentru grafice; cu ajutorul ei pot fi obţinute 32000 de culori cu un monitor de 12" şi 256 de culori cu unul de 13".

Înafară de aceasta un LC mai conţine un procesor 68020 (tact 16 MHz), 2 MBytes RAM (maxim 10 MBytes) şi un harddisk de 40 MBytes.

Cu această dotare el poate prelucra texte, fişiere şi calcula tabele de două ori mai rapid decît un Classic, dar nu atinge decît cca. 60% din puterea unui SE/30.

La intrarea audio a lui Macintosh LC poate fi cuplat un microfon (livrat de asemenea de Apple) sau de exemplu un recorder CD (Compact Disc) şi astfel pot fi digitalizate vorbire, muzică şi zgomote, pot fi prelucrate şi apoi redate. De asemenea textele şi datele pot fi însoţite de comentarii vorbite, pentru a da de exemplu indicaţii colegilor în timpul lucrului cu un program.

Deoarece fişierele de sunete necesită foarte mult spaţiu, Apple oferă un program special de comprimare, cu ajutorul acestuia pe harddisk-ul de 40 MBytes pot fi memorate înregistrări de pînă la 30 de minute.

Dezavantajul lui LC: are un singur slot liber pentru extensii hard. Consolare: deoarece acest slot (mufă de extensie) este identic cu Processor Direct Slot al lui Mac SE/30 poate fi folosită întreaga gamă de cartele de extensie disponibilă pentru SE/30.

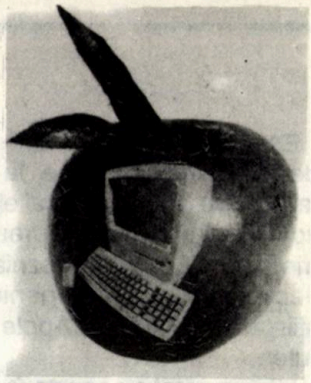
Macintosh Portable

Macintosh Portable este un calculator pentru călătorii. El dispune de o memorie de 2 MBytes şi de un harddisk de 40 MBytes. Mouse-ul tipic Mac este înlocuit cu o aşa-numită "trackball" înglobată în tastatură (o bilă prin mişcarea căreia poate fi deplasat cursorul pe ecran). Inginerii de la Apple sînt atenţi la detalii: pentru stîngaci bila poate fi montată în stînga tastaturii.

La fel de practic: cu puterea acumulatorului său, Mac Portable este economicos ca un scoţian - el poate funcţiona pînă la 10 ore independent.

Monitorul cu cristale lichide cu iluminare din fundal este de calitate ridicată: imaginile sînt construite rapid, contrastul este optim.

Cel mai mare neajuns este greutatea ridicată (7,2 kg).



Cu facilităţile sale color limitate şi viteza sa medie de calcul Macintosh LC este un calculator indicat pentru şcoală, universitate şi birou. Se recomandă pentru:

- procesare de texte
- baze de date
- calcul tabelar şi
- grafică de prezentare

A începe pentru a urca: Macintosh IIsi

Ele oferă un tempo de calcul ridicat şi o grafică excelentă - aşa-numita serie constructivă modulară (Macintosh IIsi, IIci şi IIfx). Aceste Mac-uri sînt modulare deoarece permit utilizatorului să şi le adapteze după necesităţi cu ajutorul unor plăci de extensie (module).

Cel mai mic calculator din seria modulară este Macintosh IIsi (cu monitor color de 13" începînd de la 10500 DM), calculator adus pe piaţă de Apple în octombrie.

IIsi lucrează ca şi SE/30 cu un procesor 68030. Tactul său este mai mare decît al lui SE/30 (20 MHz în loc de 16 MHz), dar trebuie să se descurce fără coprocesor şi din această cauză nu atinge decît cca. 90% din puterea acestuia.

El mai are înglobat un adaptor grafic (pînă la 256 de culori cu o rezoluţie de 384X512 pixeli), 2 MBytes RAM şi un harddisk de 40 MBytes. IIsi are un singur slot liber posibilităţile de extensie fiind astfel limitate. Totuşi utilizatorul poate utiliza fie plăci de extensie specifice pentru SE/30 fie cartele "Nubus". Trucul în cazul plăcilor de extensie: ele conţin în plus un coprocesor 68882. IIsi este ca o combinaţie între un Mac LC şi un SE/30:

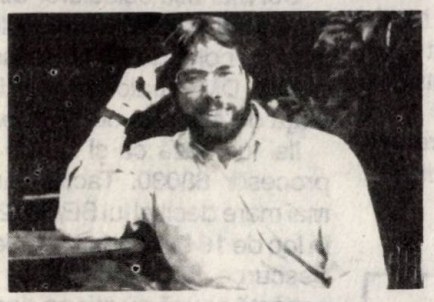
Apple Story

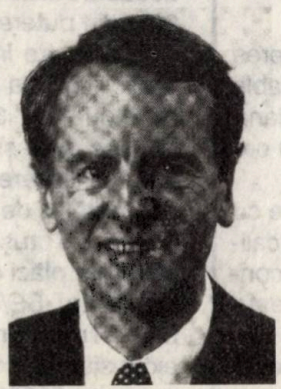
Este o poveste tipic americană "de la spălător de vase, la milionar": Steve Jobs, un fan al electronicii tînăr şi excentric, împreună cu prietenul său "Woz" Wozniak, un electronist bricoleur genial, au înfiinţat în 1976 firma Apple Computer.

Într-un garaj au construit primul calculator personal din istorie, Apple I, care a avut un succes deosebit. Cu cîştigul obţinut Jobs şi Wozniak au finanţat la sfîrşitul anilor 70 construirea calculatorului Apple II, cu care s-au impus definitiv în lumea calculatoarelor personale. Apple II a fost provizoriu ultimul mare succes pentru Apple: atît urmaşul lui Apple III, cît şi precursoarea Macintosh-urilor Lisa, n-au avut priză la public şi au adus firma Apple în pragul falimentului. Abia cu Macintosh norocul s-a reîntors.

Şeful firmei Apple Steve Jobs (azi 35 de ani şi şeful firmei Next care face concurenţă lui Apple) şi-a propus în momentul realizării modelului Macintosh să realizeze un calculator cu putere mare de calcul, confortabil în operare, fără compromisuri şi care să încapă pe masa de scris: pe scurt să construiască cel mai bun calculator personal.

După multe săptămîni de muncă de 70 de ore cu muzică rock, suc de fructe proaspăt şi jocuri pe video pentru destindere, constructorii au putut prezenta în 1984 primul calculator Macintosh: "Mac 128". Ceea ce deosebea Mac-ul de toate celelalte calculatoare era Finder: un sistem de operare cu mouse, iconuri (simboluri) şi meniuri. Modelul său: o suprafaţă utilizator pe care Jobs şi "piraţii" (aşa îşi spuneau proiectanţii înşişi) au văzut-o în Xerox PARC (PARC: Palo Alto Research Center).





Părinţii Macintosh-ului: Steve Jobs (sus) şi Stephen Wozniak (mijloc). Jos: şeful firmei Apple John Sculley

Datorită faptului că permitea o operare prietenoasă şi că era fiabilă această suprafaţă utilizator a transformat Macintosh-ul într-o legendă vie. Totuşi ea n-a putut afecta PC-urile IBM şi ale altor concurenţi printre altele datorită politicii de preţuri ridicate practicată de Apple. În cele din urmă în 1983 cînd Apple a intrat într-o criză financiară, Steve Jobs la adus la Apple pe John Sculley, pînă în acel moment manager la Pepsi, pentru a repune firma pe picioare. Sculley şi-a dat repede seama de faptul că Apple se afla în criză şi datorită stilului incoerent de conducere al lui Jobs şi l-a deposedat de putere. Cătrănit Jobs a părăsit firma Apple şi a înfiinţat o nouă firmă, Next Computer care produce staţii de lucru (workstation).

John Sculley a renunţat la politica preţurilor ridicate abia în momentul în care în ultimul an procentul din piaţă şi cîştigurile firmei Apple au început din nou să scadă rapid ca urmare a scăderilor de preţuri la calculatoarele compatibile IBM şi a apariţiei suprafeţei utilizator Windows. În plus Apple a lansat pe piaţă trei noi Mac-uri ieftine (Classic, LC, IIsi) şi a scăzut şi preţurile la modelele rămase.

Măsurile luate par să aibă succes: după cum afirmă vînzătorii de Mac-uri, mai ales Classic se vinde ca pîinea caldă. Dar nici în viitor viaţa lui Sculley nu va fi mai liniştită: noua firmă a lui Steve Jobs Next a lansat pe piaţă calculatoare care sînt mai puternice decît calculatoarele Mac II şi care costă sensibil mai puţin.

Apple s-a hotărît să riposteze concurenţei cu un nou Macintosh II cu un procesor Motorola 68040, cu tactul de 25 MHz, de îndată ce acest procesor va fi la dispoziţie în cantităţi suficient de mari.

Se mai vorbeşte şi despre un posibil Macintosh cu procesor RISC, care ar putea fi de mai multe ori mai rapid chiar decît 68040.

Pentru Macintosh Portable va exista în curînd un înlocuitor: deja la finele anului 1990 Apple a încheiat un acord cu experţii japonezi în materie de Laptop-uri Sony şi Toshiba pentru realizarea unui Macintosh Laptop care ar trebui să apară pînă la sfîrşitul acestui an. Acesta nu va fi tot la fel de puternic ca şi Portable, dar va fi mult mai uşor.

un calculator rapid, color. Acestea îl recomandă pentru utilizarea în institute de învăţămînt superior şi la birou. Se recomandă pentru:

- procesare de texte
- gestiunea bazelor de date
- calcul tabelar
- grafică de prezentare
- aplicaţii grafice semiprofesionale (de ex. publicistică).

Artistul: Macintosh IIci

Fratele mijlociu din seria modulară Mac se numeşte Macintosh IIci (preţ: 15150 DM cu monitor color 13"). El este echipat asemănător cu fratele său mai mic, dar procesorul său 68030 lucrează la un tact de 25 MHz, în plus foloseşte un coprocesor 68882 şi are o memorie de 5 MBytes.

Şi în cazul lui "Mac IIci" adaptorul grafic este înglobat pe placa de bază (max. 256 culori). El dispune însă de trei sloturi de extensie.

Cu IIci începe lumea pentru toţi cei care doresc să facă publicistică profesională. Programe de grafică puternice memorează documentele paginate pe dischete astfel încît acestea pot fi preluate şi trimise direct în tipografie.

Condiţia pentru aceasta este o putere de calcul foarte mare. Atunci cînd utilizatorul doreşte imprimarea unei imagini color, calculatorul trebuie să preia aşanumita "separare a culorilor", pentru fiecare pixel al imaginii trebuind transmise culorile constitutive.

IIsi este o alegere potrivită pentru aplicaţii în mediul universitar, la birou sau în edituri. Se recomandă pentru:

- procesare de texte
- gestiunea bazelor de date mari
- grafică de prezentare şi
- aplicaţii grafice (pictură, desen, Desktop Publishing).

Dată fiind puterea sa de calcul el poate fi utilizat, ca şi SE/30, ca server într-o reţea.

Extraterestrul: Macintosh IIfx

Nava amiral a Armadei Apple este Macintosh IIfx (preţ 22900 DM, inclusiv adaptorul grafic şi un monitor color de 13"). Prescurtarea "fx" vine de la "fast" şi "extensible" (rapid şi extensibil), deoarece IIfx este Mac-ul cel mai rapid (procesor 68030 cu tactul de 40 MHz, plus coprocesor) şi cu cele mai multe sloturi de extensie (şase).

Cu un adaptor grafic de 24 de biţi poate reprezenta simultan peste 16 milioane de culori. Macintosh IIfx este indicat pentru toate aplicaţiile pentru care era recomandat şi predecesorul său; în plus el este singurul Mac (şi probabil singurul calculator personal, incluzînd chiar şi calculatoarele compatibile IBM PC 486) care este suficient de rapid pentru următoarele categorii de aplicaţii:

- desen tehnic pentru constructori, designeri şi arhitecţi (cuvînt cheie: CAD = Computer Aided Design, proiectare cu ajutorul calculatorului);
- animaţie
- aplicaţii multimedia
- simulări

(I. M. & R. M.)

Familia Macintosh dintr-o privire

Nume produs	Mac Classic	Mac SE/30	Mac LC	Mac IIsi	Mac IIci	Mac IIfx	Mac Portable
Preţ (DM)	3000	7650	4900	10500	15150	22900	9000
Procesor Motorola							
Procesor, Tact (MHz)	68000, 8	68030, 16	68020, 16	68030, 20	68030, 25	68030, 40	68HC00, 16
Coprocesor	-	68882			68882	68882	
Memorie							
Standard/Maxim (MBytes)	2/4	2/32	2/10	2/17	5/32	4/32	2/9
Lector de dischete							
Format	3,5"	3,5"	3,5"	3,5"	3,5"	3,5"	3,5"
Capacitate (MBytes)	1,44	1,44	1,44	1,44	1,44	1,44	1,44
Harddisk							
Capacitate (MBytes)	40	40	40	40	40	80	40
Sloturi							
Tip "Processor Direct Slot"	-	1	1	(1)	-	-	1
Tip "Nubus"	-	-	-	(1)	3	6 (5 libere)	-
Interfeţe							
Imprimantă	*	*	*	*	*	*	*
SCSI	*	*	*	*	*	*	*
Modem	*	*	*	*	*	*	*
Lector de dischete extern	*	*	-	*	*	-	*
Intrare/ieşire audio	nu/mono	nu/stereo	da/mono	da/stereo	nu/stereo	nu/stereo	mono
Monitor							
Diagonală	9"	9"	12"	13"	13"	13"	9,8"
Tip	a/n, integrat	a/n integrat	color	color	color	color	a/n-LCD, integrat
Rezoluţie	512x342	512x342	512x384	640x480	640x480	640x480	640x400
Nr. nuanţelor de gri, resp. al culorilor	2	2	256	256	256	>16,7 *10^6	2
Adaptor video integrat	*	*	*	*	*		*

Joystick

Cu calculatorul te poţi şi juca, dar pentru aceasta pentru anumite jocuri ai nevoie de un joystick. În cele ce urmează o să încercăm să vă explicăm ce este şi cum funcţionează un joystick.

Jocurile pe calculator sînt un mod plăcut de petrecere a timpului liber. Pentru a putea conduce vehicule spaţiale pe ecran, a simula pilotarea unui avion sau o cursă de maşini, nu este necesar doar un anumit program ci şi o configuraţie corespunzătoare. Un accesoriu important este joystick-ul, care transformă mişcările mîinii în informaţii pentru calculator. Prin intermediul lui jucătorul ia parte la acţiunea de pe ecranul calculatorului.

Cea mai importantă componentă a joystick-ului este **maneta**. Ea este mişcată cu mîna pentru a da calculatorului comenzile de direcţionare. Dacă se doreşte mişcarea figurii proprii de pe ecran în sus, se împinge maneta în sus. Doar prin simple împingeri se pot obţine în total opt sensuri de deplasare: sus, jos, stînga, dreapta, ca şi direcţiile diagonale (diagonal dreapta sus etc.).

Atunci cînd se mişcă maneta într-o anumită direcţie, mişcarea este înregistrată de cîteva componente electronice care se află în interiorul joystick-ului. În cel mai simplu caz cu maneta se apasă tasta unui aşa-numit **microîntrerupător**. Ca şi în cazul unui întrerupător pentru lumină, se închide un circuit şi un impuls de curent slab este transmis prin intermediul **cablului de legătură** prin **mufa joystick-ului** la calculator. Prin legătură programul de joacă poate afla care sînt valorile transmise de joystick. Dacă valoarea este "zero", înseamnă că tastele de direcţie nu au fost apăsate, dacă valoarea este "unu", înseamnă că o anumită tastă a fost apăsată. Majoritatea joystick-urilor dispun pentru sesizarea direcţiei de deplasare de patru microîntrerupătoare de la care pleacă cîte un fir la calculator. Cu ajutorul lor calculatorul poate afla şi dacă maneta a fost împinsă, de ex., diagonal. stînga jos. În acest caz vor fi pe "unu" simultan valorile pentru "jos" şi "stînga".

Joystick-urile cu microîntrerupătoare sînt foarte convenabile (performanţe/preţ) şi din acest motiv sînt utilizate de o mulţime de jocuri pentru HC-uri (Home Computer). Ele pot indica doar direcţiile de deplasare, dar nu şi cît de departe a fost împinsă maneta într-o anumită direcţie. Pentru aceasta sînt necesare aşa-numitele joystick-uri analogice, care ştiu deosebi şi situaţii intermediare cum ar fi "50% dreapta" sau "95% dreapta". În timp ce celelalte joystick-uri asemenea unui întrerupător pentru lumină nu cunosc decît poziţiile "închis" şi "deschis", modelele analogice acţionează ca un reostat care permite ca lumina să scadă sau să crească în trepte. Avantajul: acolo unde contează sensibilitatea degetelor, de ex. la simularea zborului cu un avion cu reacţie, se poate conduce mai exact. Joystick-urile cu microîntrerupătoare sînt denumite, pentru deosebire, "joystick-uri digitale".

Joystick-urile digitale utilizează în locul microîntrerupătoarelor două "potenţiometre". Cu cît este deplasat cursorul unui potenţiometru, prin intermediul manetei, mai mult într-o anumită direcţie, cu atît mai mare sau mică este doza de curent care este transmisă prin intermediul cablului de legătură la aşa-numitul "convertor analog-digital", care transformă impulsurile de curent în valori digitale. În acest caz nu se face însă distincţie doar între "zero" şi "unu" ci pot fi deosebite pînă la 64 de valori intermediare. Valoarea "zero" semnifică, de ex., "la stînga de tot", iar valoarea "63" înseamnă "la dreapta de tot". Acelaşi lucru este valabil şi pentru cel de-al doilea potenţiometru, care semnalează poziţiile de sus pînă jos. Atunci cînd maneta este la mijloc ambele potenţiometre trebuie să indice valoarea "31".

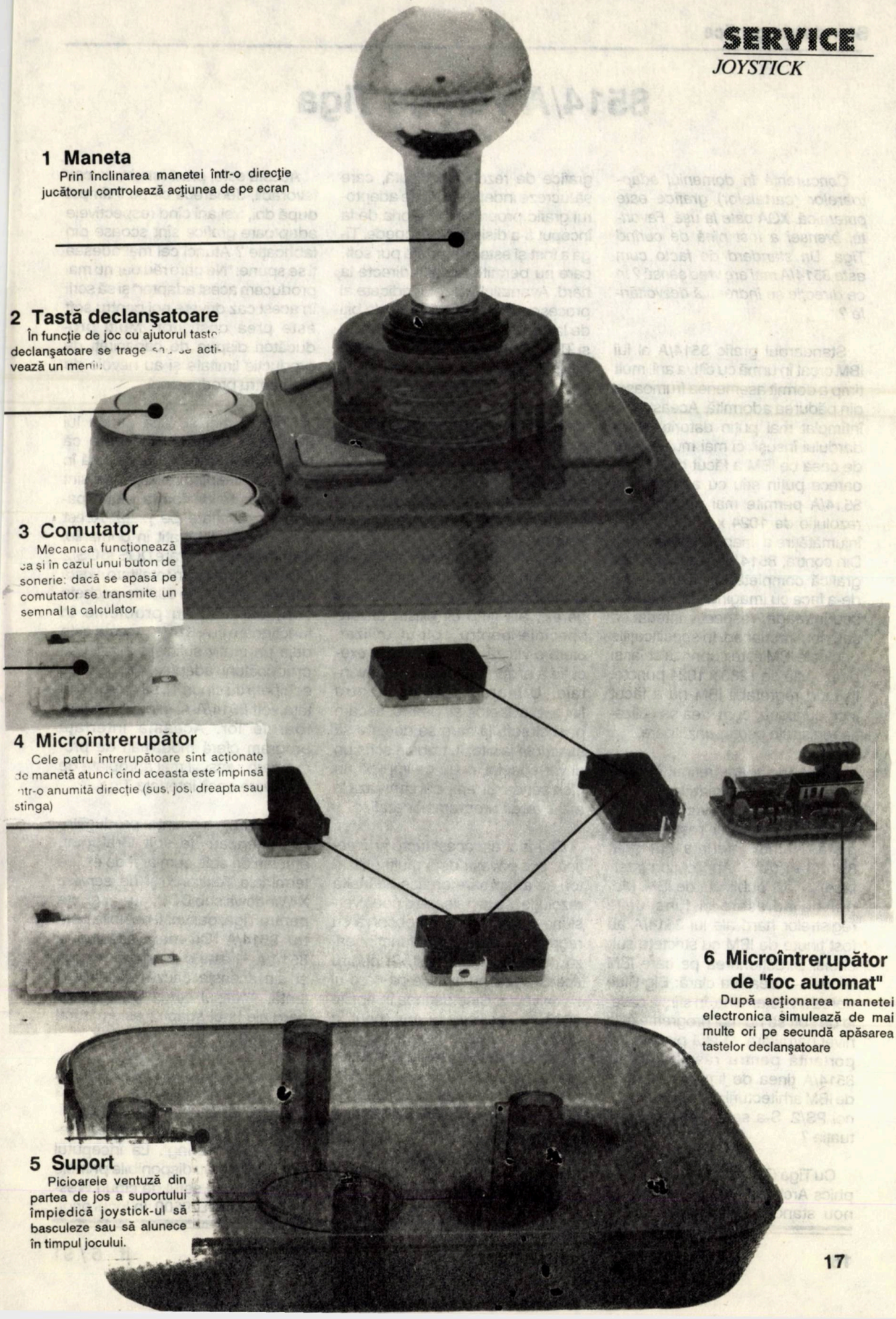
Joystick-urile analogice sînt utilizate mai ales pentru PC-uri. Acestea au nevoie în plus de un aşa-numit "port pentru jocuri" (Preţ: 50 DM plus 50 DM pentru joystick; joystick-urile digitale costă între 10 şi 40 DM). Acesta este de fapt o placă de extensie care conţine o interfaţă pentru joystick. Majoritatea jocurilor se pot lipsi însă de un joystick, operarea putîndu-se face de la tastatură sau cu un mouse.

Înafara manetei de comandă o altă componentă importantă a joystick-ului o reprezintă **tastele declanşatoare** (trăgaciul). Prin intermediul lor jucătorul poate transmite alte comenzi calculatorului, asemănător ca prin intermediul tastelor de pe mouse. Prin intermediul tastelor jucătorul poate trage asupra "inamicilor" sau poate selecta o opţiune dintr-un meniu.

Joystick-urile bune au şi o **tastă de "foc automat"**, care simulează apăsarea repetată a tastelor. Jucătorul se poate concentra astfel doar asupra conducerii.

La un joystick sînt foarte importante şi **picioarele de sprijin**, care să dea o stabilitate joystick-ului în timpul jocului. Din acest motiv multe modele dispun de picioare ventuză.

(R.M.)



1 Maneta

Prin înclinarea manetei într-o direcţie jucătorul controlează acţiunea de pe ecran

2 Tastă declanşatoare

În funcţie de joc cu ajutorul tastei declanşatoare se trage sau se activează un meniu

3 Comutator

Mecanica funcţionează ca şi în cazul unui buton de sonerie: dacă se apasă pe comutator se transmite un semnal la calculator

4 Microîntrerupător

Cele patru întrerupătoare sînt acţionate de manetă atunci cînd aceasta este împinsă într-o anumită direcţie (sus, jos, dreapta sau stînga)

6 Microîntrerupător de "foc automat"

După acţionarea manetei electronica simulează de mai multe ori pe secundă apăsarea tastelor declanşatoare

5 Suport

Picioarele ventuză din partea de jos a suportului împiedică joystick-ul să basculeze sau să alunece în timpul jocului.

8514/A versus Tiga

Concurenţa în domeniul adaptoarelor (cartelelor) grafice este puternică. XGA bate la uşă. Favoritul branşei a fost pînă de curînd Tiga. Un standard de facto cum este 8514/A mai are vreo şansă ? În ce direcţie se îndreaptă dezvoltările ?

Standardul grafic 8514/A al lui IBM creat în urmă cu cîţiva ani, mult timp a dormit asemenea frumoasei din pădurea adormită. Aceasta s-a întîmplat mai puţin datorită standardului însuşi, ci mai mult a ţinut de ceea ce IBM a făcut din el. Deoarece puţin ştiu cu adevărat că 8514/A permite mai mult decît o rezoluţie de 1024 x 768 în modul înjumătăţire a imaginii (interlaced). Din contră, 8514/A este o interfaţă grafică completă şi nu are nimic de-a face cu imaginea înjumătăţită sau întreagă, respectiv interlaced sau non-interlaced. În specificaţiile originale IBM este cuprinsă chiar şi o rezoluţie de 1280 x 1024 puncte. În mod regretabil IBM nu a făcut încă cunoscut cum vrea să utilizeze registrele corespunzătoare.

IBM a fost foarte reţinut în ceea ce priveşte informaţiile despre 8514/A, ceea ce a avut ca efect o acceptanţă relativ restrînsă pe piaţă. Numai interfaţa pur soft, 8514/A-AI (AI = Application Interface), a fost publicată de IBM. Modul de adresare şi funcţiunile registrelor hard ale lui 8514/A au fost ţinute de IBM cu stricteţe sub cheie. Direcţionarea pe care IBM dorea s-o dea era clară: Big Blue dorea să determine, în sfîrşit, casele de soft să nu mai programeze la nivel hardware. O altă piedică importantă pentru răspîndirea lui 8514/A ţinea de limitările impuse de IBM arhitecturii microcanal a seriei PS/2. S-a schimbat cumva situaţia ?

Cu Tiga (Texas Instruments Graphics Architecture) s-a realizat un nou standard pentru adaptoare grafice de rezoluţie ridicată, care să lucreze independent de adaptorul grafic propriu zis. Euforia de la început s-a disipat însă repede. Tiga a fost şi este o interfaţă pur soft, care nu permite accese directe la hard. Avantajul vitezelor ridicate al procesoarelor grafice pe 32 de biţi de la Texas Instruments (TI-34010 şi TI-34020) se pierde din această cauză.

Acest lucru a fost observat de către cîţiva dintre producătorii de adaptoare grafice cu procesoare TI, şi aceştia nu mai oferă chiar nici o interfaţă Tiga. Să scrii drivere unitare pentru astfel de adaptoare este imposibil deoarece fiecare adaptor grafic are un alt design hard. Astfel există adaptoare care sprijină în mod special funcţiile pan şi zoom sau alte caracteristici, care de ex., atît timp cît există drivere speciale pentru soft-ul utilizat, oferă o viteză mai ridicată de execuţie a programelor sau alte avantaje. Urmarea este că pentru fiecare adaptor şi pentru fiecare pachet soft, la care se doreşte să se umble la viteză, trebuie scris un driver special, ceea ce implică un efort serios. Şi tempoul contează la majoritatea programelor grafice.

Ce efect are acest lucru în practică, pot povesti deja mulţi utilizatori de adaptoare grafice de înaltă rezoluţie: se instalează o nouă versiune a unui soft şi se observă cu regret că programul nu funcţionează pe PC-ul disponibil. Şi pentru aceasta nici nu trebuie ca hard-ul să fie vechi. Cine deţinea în aprilie 1990, de ex., ce era mai "nou" în domeniul adaptoarelor grafice, s-a trezit nu mai departe decît în iunie, la instalarea lui Windows 3.0, că se loveşte de aceleaşi probleme de care se lovise şi cu un an înainte la actualizarea versiunii Windows 2.03 cu mai noua versiune 2.1. Trebuiau aşteptate drivere noi, şi aceasta de regulă durează mai multe luni.

Acesta este, privit relativ, un caz favorabil, deoarece ce se întîmplă după doi, trei ani cînd respectivele adaptoare grafice sînt scoase din fabricaţie ? Atunci cel mai adesea ţi se spune: "Ne pare rău dar nu mai producem acest adaptor" şi să scrii în acest caz drivere noi pentru soft este prea costisitor. Mulţi producători dispun de capacităţi de producţie limitate şi au nevoie de ele pentru produse noi.

Tocmai aici se află şansa lui 8514/A. Deoarece cu toate că adaptoarele 8514/A nu au pînă în prezent o răspîndire largă, ele sînt recunoscute de foarte multe pachete soft aflate pe piaţă. Acest lucru este valabil atît în domeniul sistemelor de operare DOS şi OS/2 cît şi pentru Unix. În totalitate există astăzi mai mult de 300 de pachete soft, care nu au probleme în funcţionare cu 8514/A. Acesta este deja un motiv suficient pentru ca producătorii adaptoarelor grafice echipate cu cipuri TI, să ofere interfaţa soft 8514/A-AI şi pentru adaptoarele lor. Această interfaţă program oferă modalităţi de programare sensibil mai apropiate de hardware decît interfaţa Tiga.

După cum a anunţat producătorul american de soft Grafpoint, emulări de soft, cum ar fi de ex. de terminale Tektronix şi de servere X-Windows sub DOS, nu vor exista pentru Tiga, dar vor fi posibile pentru 8514/A. Cu toate acestea şi 8514/A-AI este o interfaţă pur soft şi din această cauză este relativ lentă. Accesul direct la registrele hard ale unui adaptor este natural mai rapid.

În 1989 producătorului american Western Digital i-a reuşit să citească conţinutul registrelor cipului IBM-8514/A prin tehnica "Reverse-Engineering". La începutul acestui an erau disponibile primele adaptoare cu setul de cipuri Western Digital WD-9500. În locul origi-

nalelor au început să apară clone interesante ca preţ. Şi aceste clone sînt disponibile nu numai pentru microcanal ci şi pentru magistralele normale ale AT-urilor. Copiile au o rezoluţie de 1024 x 768 puncte la 16 şi respectiv 256 de culori şi lucrează în plus în modul imagine întreagă (non-interlaced).

Această situaţie - după ce registrele hard ale lui 8514/A au devenit cunoscute - face acest adaptor încă şi mai interesant pentru ca producătorii mari de soft să-l trateze în pachetele lor. Microsoft a făcut acest lucru cu Windows 3.0, acelaşi lucru făcîndu-l şi Autodesk cu Autocad 11.0. Tocmai succesul uriaş al lui Windows 3.0 face ca 8514/A să apară ca un standard ideal pentru adaptoare grafice. Se poate lucra cu el fără probleme nu numai cu bogata paletă de aplicaţii Windows, ci şi cu programe CAD cum ar fi Autocad, PC-Draft sau cu soft-uri asemănătoare.

Nici o grijă pentru soft

Şi în domeniul Unix, 8514/A se scrie cu litere mari. X-Windows funcţionează pe 8514/A, acelaşi lucru întîmplîndu-se şi cu programele producătorilor Interactive şi SCO (Santa Cruz Operation). Servere DOS pentru X-Windows cu acest standard vor fi disponibile cel mai tîrziu în august 1991.

Aceste produse se află momentan în faza de beta-testare, deci sînt aproape de lansarea pe piaţă. Criteriul important şi hotărîtor al întregii situaţii este că utilizatorul nu trebuie să-şi facă probleme pentru modul în care va fi sprijinit soft-ul. Pentru aceasta se angajează în competiţie tot mai mulţi producători de adaptoare grafice.

Efortul de elaborare al unui driver per adaptor grafic şi pachet soft se situează între 20000 şi 100000 DM. O investiţie care într-un fel sau altul trebuie plătită de utilizatori şi nu prezintă nici o garanţie de viitor.

Testele comparative între Tiga şi clonele 8514/A au produs o surpriză în favoarea lui 8514/A. Adaptorul 8514/A echipat cu un set de cipuri WD-9500 la un tact de 60 MHz a fost în medie de trei ori mai rapid decît un adaptor echipat cu procesorul TI-34010, la acelaşi tact.

Clonele sosesc

Alţi producători se află deja în preajma lansării pe piaţă a unor seturi de cipuri 8514/A. Astfel un producător american de adaptoare grafice a anunţat pentru această primăvară un clon 8514/A, cu un tact de 80 MHz, bazat pe cipuri de la firma Chips & Technologies (C&T). Cu acestea ar trebui să se atingă o viteză care să echivaleze puterea adaptoarelor grafice echipate cu procesorul grafic TI-34020, dar la preţuri sensibil mai scăzute.

Preţul final de utilizare al acestui adaptor se va situa sub 1800 DM. Acesta ar fi cu cca. 60% mai ieftin decît un adaptor cu TI-34020. Pentru acesta ar fi posibilă o rezoluţie de 1024 x 768 la 256 de culori şi în cazul unei echipări corespunzătoare chiar de 1280 x 1024 la 256 de culori.

Premisa de funcţionare a unui adaptor grafic 8514/A este utilizarea unui monitor multi-frecvenţă, deoarece unele soft-uri accesează direct registrele hard. Aceasta poate însemna în cazuri concrete o rezoluţie ecran de 640 x 480 puncte, 800 x 600 puncte, sau pînă la 1024 x 768 interlaced, respectiv 1024 x 768 non-interlaced. Aceasta este dependentă doar de soft şi nu poate fi influenţată din afară.

Totuşi nu este cunoscut nici un soft care să utilizeze rezoluţia de 800 x 600 după specificaţiile 8514/A, iar multe programe lucrează în modul 1024 x 768 non-interlaced.

În dezvoltarea tehnică a adaptoarelor grafice sînt de aşteptat cîteva noutăţi interesante. Astfel vor exista adaptoare cu seturi de cipuri VGA "on-board". Prin două ieşiri separate pentru semnale VGA şi 8514/A şi o "Pass-Through" (trecere) internă a semnalelor VGA peste ieşirea de înaltă rezoluţie, va exista posibilitatea de a utiliza unul sau două monitoare cu un singur adaptor grafic. Avantajul imediat: pentru întregul sistem grafic va fi utilizat un singur conector (slot) la magistrală.

Şi la capitolul ergonomie adaptoarele 8514/A au un avans: monitoarele cu o frecvenţă de împrospătare a imaginii de 70 Hz aparţin deja standardului. Elaborarea unor adaptoare cu o frecvenţă pe verticală peste 80 de Hz este perfectată. Problema va fi doar să se găsească acele monitoare care să poată lucra la o astfel de frecvenţă.

Pînă acum i-a reuşit unui singur producător "să-i sufle" IBM-ului un standard grafic de pe piaţă, este vorba despre Hercules. Aceasta s-a întîmplat însă numai pentru un timp deoarece astăzi Hercules se retrage: noile adaptoare grafice Hercules sînt compatibile cu toate standardele grafice IBM, inclusiv 8514/A.

Nu numai în domeniul impunerii unor standarde grafice a fost IBM cel mai mare. Egal, cît de bune sau cît de proaste au fost aceste standarde, toţi ceilalţi producători au urmat, mai devreme sau mai tîrziu, direcţia de marş a uriaşului albastru. De îndată ce avantajele celui mai bun standard, lansat pînă acum de IBM, vor deveni cunoscute, nu mai există nici o îndoială: cu sau fără sprijinul lui IBM 8514/A se va impune alături de XGA.

(R. M.)

Merită să mizezi pe XGA ?

Cu anunţarea de către IBM a lui XGA s-a deschis un nou capitol în problema standardelor grafice, o sentinţă rămîne însă: din punct de vedere tehnic sînt puţine noutăţi de semnalat. Clienţii şi distribuitorii sînt încurcaţi. Ce este XGA, ce avantaje aduc specificaţiile ? Se vor impune ?

IBM a anunţat un nou standard - şi întreaga branşă este tulburată. Extended Graphics Array (XGA) este conceptul care înfierbîntă multe sentimente în şi aşa destul de agitata lume a afacerilor cu grafică. "Interfaţa grafică XGA este o implementare busmaster, care oferă o nouă dimensiune în grafică", afirmă bineînţeles Big Blue. Totuşi părerile celor din branşă sînt discordante.

În urma unui chestionar fulger dat de revista PC Magazin mai multor distribuitori şi producători de adaptoare grafice a rezultat că: unii livrează în continuare din stocurile de adaptoare 8514/A şi consideră că acestea au încă şanse bune să se menţină, în timp ce alţii sînt entuziasmaţi de XGA. Un comentariu exemplificator va lămuri îndoiala unor cunoscători ai branşei: "marfă veche în amabalaj nou, atrăgător - foarte iscusit din parte lui IBM".

Cum poate fi înţeles acest lucru ? Vechi este XGA deoarece rezoluţia de 1024 x 768 dpi (dot per inch) exista deja de mult, şi chiar din casa IBM-ului (8514/A). Nou este ambalajul: XGA este compatibil soft cu VGA, 8514/A, Windows, Presentation Manager şi Autocad. Aplicaţiile corespunzătoare trebuie deci să funcţioneze fără probleme.

Oferta de standarde în segmentul de piaţă al rezoluţiilor oferite de VGA Plus, Super-VGA şi 8514/A era încîlcită. Texas Instruments proclamase interfaţa soft Tiga, pentru adaptoare grafice de rezoluţie ridicată echipate cu procesoare TI din familia TMS 340, ca "inteligentă". Vesa specificase un standard propriu, iar producători ca Miro sau Spea pătrunseseră pe piaţă cu rezoluţii de 2048 x 2048 pixeli şi cu drivere proprii. IBM s-a ţinut cu tărie de 8514/A şi a aşteptat să vadă dacă va exista necesitatea unor rezoluţii mai mari sau a mai multor culori.

O lipsă cunoscută era că introducerea unei plăci grafice de rezoluţie ridicată, cel mai adesea scumpă, era dependentă de existenţa driverelor corespunzătoare pentru soft. Dacă XGA se va impune, atunci se va termina cu aşteptarea după drivere speciale, parţial specifice chiar adaptorului utilizat. Istoria PC-urilor a arătat de mai multe ori că standardele stabilite de IBM s-au impus cu repeziciune. În cazul XGA acest fenomen s-ar putea repeta.

Ce avantaje oferă XGA ? Big Blue promite o creştere sensibilă a puterii sistemelor cu microcanal, deoarece adaptorul ca busmaster preia anumite funcţiuni ale procesorului principal degrevîndul pe acesta şi lăsîndu-l să se ocupe între timp de alte probleme. IBM vorbeşte despre adaptorul XGA ca despre un subsistem grafic. Pînă acum este cunoscut că adaptorul lucrează cu un procesor RISC (Reduced Instruction Set Computer), al cărui set redus de instrucţiuni permite o creştere a vitezei. Suplimentar, utilizînd un adaptor XGA în modul standard VGA sînt posibile mai multe culori la o aceeaşi rezoluţie.

XGA ca busmaster

Extended Graphics Array a fost folosit mai întîi în sistemele microcanal PS/2 90 şi 95. În modelul 90 XGA este integrat pe placa de bază ca interfaţă monitor. În 95 XP 486, XGA este prevăzut standard ca adaptor busmaster. Acelaşi adaptor grafic XGA poate fi instalat, la dorinţă, în toate modelele PS/2 cu procesoare 80386SX, 80386 şi 80486. Într-un sistem PS/2 vor fi sprijinite mai multe astfel de adaptoare. Alte caracteristici: setul tehnologic VRAM (Video Random Access Memory), un cursor hard propriu şi drivere pentru aplicaţii CAD (Computer Aided Design) speciale.

Ce rezoluţii sînt posibile ? Posesorii de modele PS/2 sau de PC-uri cu microcanal pot fi liniştiţi, nu au nevoie de un nou monitor. XGA se bazează în mare pe standardele 8514/A, şi respectiv pe Super VGA şi partajează cu acestea rezoluţia de 1024 x 768 pixeli la 256 de culori. În modul VGA, subsistemul de regiştri XGA este compatibil cu standardul VGA. În mod text XGA permite afişarea pe ecran a 132 de caractere, a cîte 8 pixeli. În cazul rezoluţiei de 640 x 480 pixeli pot fi reprezentate 65536 de culori.

Branşa a privit pozitiv intrarea lui IBM în VESA (Video Electronics Standard Association). Acest pas indică cu claritate renunţarea lui IBM la rolul "călăreţului singuratec" şi lasă speranţe pentru consolidarea pieţei de grafică. Cunoscătorii afirmă că deşi specificaţiile XGA au fost date publicităţii vor trece 9 pînă la 12 luni pînă cînd vor putea fi dezvoltate adaptoare noi. Greutăţile vor fi produse mai ales de interfaţa de 32 de biţi, ca şi de cipurile care sînt mai complexe la XGA decît la VGA. După IBM este posibilă o creştere a rezoluţiei de pînă la 4096 x 4096 pixeli.

Dataquest a criticat totuşi faptul că acest lucru nu este definit în mod clar, deoarece spaţiul stabilit de IBM ar putea duce la incompatibilităţi. Un lucru este clar însă: IBM doreşte să sprijine producătorii de adaptoare. Din această cauză XGA are şanse bune de a se impune. Doi "stropi de vermut" există totuşi: concurenţa din propria casă, 8514/A, care tocmai a început să se impună, şi adaptorul Tiga care pînă acum a fost favorizat.

(R. M.)

Standarde grafice

Anul introducerii	Standard grafic	Rezoluţie	Mod	Culori	Matricea de pixeli / caracter	Moduri compatibile hard	Frecvenţa verticală (Hz)	Frecvenţa orizontală (KHz)
1981	**MDA** (Color Graphics Adapter)	720x350	Text	1	9x14	-	50	18.43
1981	**CGA** (Color Graphics Adapter)	640x200	Text	16	8x8	-	60	15.75
		320x200	Text	16	8x8	-	60	15.75
		160x200	Grafic	16	-	-	60	15.75
		320x200	Grafic	4	-	-	60	15.75
		640x200	Grafic	2	-	-	60	15.75
1982	**MGA** (Hercules Monochrome Graphics Adapter)	720x350	Text	1	9x14	MDA	50	18.10
		720x348	Grafic	1	-	MDA	50	18.10
1984	**EGA** (Enhanced Graphics Adapter)	640x350	Text	16	8x14	CGA,MDA	60	21.85
		720x350	Text	4	9x14	CGA,MDA	60	21.85
		640x350	Grafic	16	-	CGA,MDA	60	21.85
		320x200	Grafic	16	-	CGA,MDA	60	21.85
		640x200	Grafic	16	-	CGA,MDA	60	21.85
		640x350	Grafic	16	-	CGA,MDA	60	21.85
1984	**PGA** (Professional Graphics Adapter)	640x480	Grafic	256	-	CGA	60	30.50
1987	**VGA** (Video Graphics Adapter)	720x400	Text	16	9x16	CGA,EGA	70	31.50
		360x400	Text	16	9x16	CGA,EGA	70	31.50
		640x480	Grafic	16	-	CGA,EGA	60	31.50
		640x480	Grafic	2	-	CGA,EGA	60	31.50
		320x200	Grafic	256	-	CGA,EGA	70	31.50
1987	**MCGA** (Memory Controller Gate Array)	320x400	Text	4	8x16	CGA,EGA	70	31.50
		640x400	Text	2	8x16	CGA,EGA	70	31.50
		640x480	Grafic	2	-	CGA,EGA	60	31.50
		320x200	Grafic	256	-	CGA,EGA	70	31.50
1989	**Super VGA** (recomandări VESA)	800x600	Grafic	16	-	VGA,CGA, EGA	56, 60 sau 72	35.00, 37.60 sau 48.00
1987	**8514/A**	1024x768	Grafic	16	-	VGA şi compatibilele în jos	43.48	35.52
		640x480	Grafic	256	-		43.48	35.52
		1024x768	Grafic	256	-		43.48	35.52
1990	**XGA** (Extended Graphics Array)	640x480	Grafic	256	-	VGA	43.48	35.52
		1024x768	Grafic	256	-	VGA	43.48	35.52
		640x480	Grafic	65536	-	VGA	43.48	35.52
		1056x400	Text	16	8x16	VGA	43.48	35.52

(Sursa: PC Magazin 4/91)

Evaluarea performanţelor harddisk-urilor

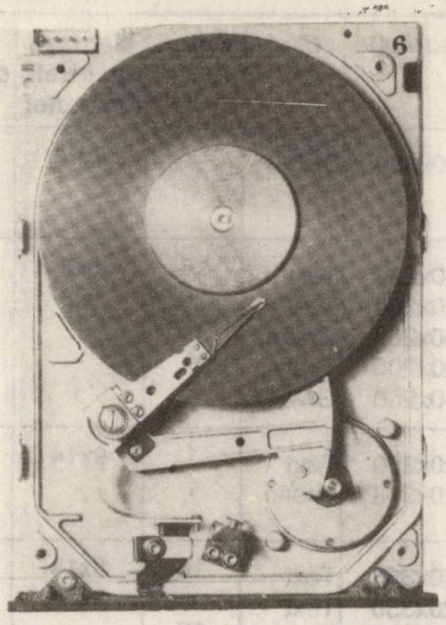
Cumpărătorul unui PC, chiar dacă de cele mai multe ori preferă soluţia "de-a gata", din ce în ce mai des are posibilitatea opţiunii la configurarea sistemului cu care va lucra. Cîteva criterii de decizie pentru alegerea harddisk-ului sînt explicitate în acest articol.

Trei sînt criteriile cele mai importante: preţul, capacitatea şi "performanţa" unităţii de disc, "performanţa" fiind o sinteză a mai mulţi parametri, ca de exemplu timpul de răspuns (acces), viteza de transfer, tratarea erorilor, fiabilitatea ş.a.m.d.

Sistemele de operare precum VMS, OS, Unix ca şi toate programele moderne de aplicaţii (proiectare 3D, sisteme complexe de baze de date) lucrează foarte intens cu memoria externă - unităţile de disc. Deseori, la sistemele multiutilizator, motivul principal pentru încetineala exasperantă a calculatorului nu-l constituie puterea insuficientă a unităţii centrale, ci performanţa redusă a discului. Fenomenul survine în special la aplicaţii intensive în privinţa datelor, atît comerciale cît şi ştiinţifice. (Mie mi s-a întîmplat chiar pe un VAX, cînd proiectam cu editorul grafic "Princess").

Costul

Depinde desigur de firma producătoare şi tipul unităţii de harddisk (vezi tabelul). Trebuie avută în vedere minimizarea costului total, nu pur şi simplu cumpărarea celui mai ieftin disc disponibil la capacitatea dată. Trebuiesc avute în vedere mai multe considerente: interfaţa, controller-ul şi funcţiile suplimentare ale acestuia, compatibilitatea cu restul echipamentului (nu doar ca interfaţă sau magistrală!), funcţiile integrate (precum Cache Look-Ahead sau tratarea "transparentă", în timp real, a defectelor), posibilităţile de extindere ulterioară şi fiabilitatea. Mai ales în condiţiile noastre, unde depanarea ridică probleme, ar trebui să ezităm a cumpăra un harddisk garantat doar 6 luni sau un an şi cu MTBF (Mean Time Between Failures - timpul mediu de defectare) nedepăşind 25.000 ore, chiar dacă-i substanţial mai ieftin.



Totodată, nu trebuie exagerat - pentru aplicaţiile uzuale, Hewlett Packard oferă discuri performante, cu MTBF de 150 şi chiar 175.000 ore, iar Seagate dă la modelele de vîrf 150.000 ore în mediul unui birou obişnuit şi 250.000 în sală calculator climatizată (clasa A). Ceea ce înseamnă 30 de ani la exploatarea non-stop în 3 schimburi şi peste 100 de ani la utilizarea "casnică". Chiar în România, durata de viaţă morală a unui sistem nu depăşeşte 25 de ani!

Capacitatea

Ce semnifică cele două cifre din tabel ? Capacitatea brută este capacitatea totală, fizică, produsul dintre nr. total de piste x nr. de octeţi ce pot fi scrişi pe o pistă. Capacitatea netă este cea utilă şi este semnificativ mai mică - nr. octeţi/sector x nr. sectoare/pistă x nr. piste/faţă disc x nr. feţe disc. Diferenţa se datorează în principal spaţiului dintre sectoare, informaţiilor adiţionale înscrise în antetul şi "coada" fiecărui sector, spaţiului (eventual) rezervat pentru compensarea erorilor. Factorul de corecţie poate fi de 10-20%. Unele firme dau capacitatea brută, fiindcă livrează discul neformatat, deci utilizatorul poate să şi-l organizeze cum doreşte, altele specifică ambele valori sau numai pe ultima - capacitatea netă - dacă livrează discul complet formatat.

Ce capacitate este necesară?

Cel mai simplu este de estimat în cazul aplicaţiilor de tip prelucrare text: o pagină A4 este echivalentă cu circa 3 ko, deci pe un disc de 40 Mo se pot înmagazina circa 12.000 de pagini de text dactilografiat. De fapt, valoarea reală poate fi considerabil mai mică: informaţia în realitate nu-i compactă, ci cu mai multe "auxiliare", versiuni anterioare, fişiere provizorii etc. Sistemul de lucru ocupă de asemenea spaţiu pe disc. Programele de procesare de texte ocupă adesea 2 Mo sau mai mult, iar aplicaţii specifice de editare - tehnoredactare, precum Ventura Publisher sau Pagemaker, au nevoie de 3 - 10 Mo sau chiar mai mult, cînd se lucrează cu imprimante laser sau postscript.

Ţinînd cont şi de versiunile multiple, chiar şi un disc de capacitate mare se umple repede. În afară de aceasta, totdeauna trebuie rezervat loc pentru date neprevăzute: altfel, se poate ajunge la situaţii de blocaj.

Trebuie estimat optimul compromisului capacitate - preţ: discul să nu fie supradimensionat, dar nici subdimensionat.

Uneori, cumpărarea unui harddisk fără rezerve prea mari de spaţiu şi înlocuirea lui ulterioară cu unul de capacitate mai mare poate fi o soluţie mai ieftină decît cumpărarea directă a modelului mai scump, dacă timpul dintre achiziţia iniţială şi schimb nu-i prea mare. Deci, se va avea în vedere un harddisk suficient pentru următorii

1 - 3 ani.

Observaţie: relaţia capacitate - preţ nu e liniară; modelele mai mari au costul specific mai redus. În plus, şi timpul de acces este mai mic decît la modelele mai mici, ieftine.

Performanţa

Un criteriu deosebit de important pentru performanţă este timpul în care (în medie statistică) subsistemul de disc duce la îndeplinire o comandă de scriere sau citire a datelor. Acest timp se însumează cu timpii necesari pentru diversele operaţii pregătitoare pentru execuţia comenzii. Pentru evaluarea performanţei harddiskului nu este relevant timpul de overhead ("regie") al sistemului de operare şi cel al controller-ului de disc, deoarece aceştia sînt relativ independenţi de tipul unităţii de disc folosite. Pentru simplificare, putem considera - în primă aproximaţie - că unitatea de disc este legată direct la calculator, care-i sursa de date. Cînd unitatea de disc primeşte o comandă de citire sau scriere, trebuie ca mai întîi blocul capetelor să se poziţioneze pe cilindrul unde există primul bloc de transferat. Acest timp se numeşte timp de poziţionare. Mărimea sa depinde evident de locul unde se face poziţionarea, astfel că se ia, în mod normal, timpul în care capetele parcurg 1/3 din lungimea maximă (30% din cursă). Se poate demonstra matematic că, la blocuri de date necorelate, avem statistic L poziţ. medie = 1/3 L poziţ. max., relaţia fiind valabilă şi pentru timpi. Valori extreme: <= 10 ms la discurile cele mai rapide, 20 - 30 ms în medie, 100 ms sau chiar mai mult la dischete sau discuri optice. Pentru siguranţă, firmele serioase dau, în afara timpului mediu de poziţionare, şi pe cel maxim (pentru 100% din cursa capetelor). Unele discuri mai necesită un timp de poziţionare mecanică, pentru stabilizarea la locaţia respectivă. Acest timp de stabilizare (settling) se specifică de obicei doar la unităţile "mici" de discuri, de 5 1/4" diametru.

După poziţionare, trebuie aşteptat pînă ce suprafaţa discului, care se roteşte, aduce blocul de date dorit în dreptul capetelor de citire/scriere. Acesta este timpul de latenţă. Majoritatea unităţilor de disc rotesc discurile cu 3600 rpm (rotaţii pe minut), astfel că timpul necesar pentru o jumătate de rotaţie este de cca 8ms. Fiind egal la toate discurile, nu-i un criteriu pentru departajare. Dar atenţie! Unităţile de disc contemporane care vor să asigure performanţe superioare utilizează turaţii mai mari (lucru imposibil la discurile vechi, de diametru mare, >= 8"). Lucrînd la peste 4000 sau chiar 5600 r.p.m., timpul de latenţă scade la cca 5 ms - şi atunci poate fi criteriu de selecţie.

Urmează transferul datelor de pe disc la controller sau invers. Dacă nu se transferă cantităţi uriaşe de date la o singură cerere de citire sau scriere, şi acest timp poate fi dat factor comun şi neglijat. Sistemele de operare precum VMS şi OS, ca şi majoritatea aplicaţiilor (sisteme de baze de date, procesare de text, etc.) transferă în general puţine blocuri de date la o cerere, tipic numai 4-8 blocuri/cerere de citire/scriere, astfel că timpul de transmisie pentru respectiva cerere este de 1-3 ms la majoritatea unităţilor de disc.

Deci, în mod normal, timpul total de acces (Zugriff în germană) este suma timpilor de: poziţionare, stabilizare, latenţă, şi transfer, deci este cu un ordin de mărime superior timpului de transfer, care şi el poate fi neglijat.

De altfel, o viteză foarte mare de transfer a datelor între disc şi controller nu-i utilă dacă acesta din urmă nu poate să debiteze datele pe magistrală la fel de repede.

De aceea, actualmente majoritatea controllerelor de harddisk-uri contemporane au buffere (memorii-tampon) SMB, de adaptare a vitezelor. Sarcina SMB este de a "potrivi" viteza de transfer ridicată dintre disk şi controller cu cea mai lentă, dintre controller şi memoria calculatorului, memorînd temporar datele pînă ce sînt transferate. Altfel pentru a evita riscul pierderii de date, transferul de pe disc la controller ar fi trebuit să se facă mai lent decît posibilităţile reale, intrinseci.

Deci, rata de transfer a datelor între disk şi controller este edificatoare mai curînd pentru performanţele şi puterea întregului sistem, decît ale unităţii de disc propriu-zise.

Mic dicţionar - harddisk-uri

Tratarea automată a defectelor

Funcţia de tratare automată a defectelor ("defect management") este integrată, de exemplu, la seria ProDrive a firmei Quantum. Este un sistem de tratare automată a defectelor integrat sub formă de firmware în unitatea de disc. Se elimină astfel necesitatea ca utilizatorul să ridice hărţi ale defectelor (clasica tabelă cu "bad blocks") şi să scrie soft utilitar dedicat evitării folosirii acestor zone.

La fabrică, suprafeţele de date sînt baleiate pentru detectarea sectoarelor defecte. Orice sector defect este dez-alocat înainte de a expedia unitatea de disc. Se realizează automat o hartă inline (ILDM) a defectelor, care conţine locaţiile tuturor sectoarelor cu defecte găsite pe disc. Pentru protecţie la erori, harta este duplicată pe mai multe piste. În plus, în timpul utilizării normale, unitatea de harddisk continuă să compenseze (detecteze + ocolească) orice noi sectoare cu defecte apărute pe disc. Toate defectele găsite la utilizator sînt cartate direct în sectoarele de înlocuire. Toată această tratare a defectelor este complet transparentă pentru utilizator, care n-are nici o grijă !

Valori tipice: rata de transfer "internă" poate ajunge la 10 şi chiar 28 Mo/s, în timp ce transferul "extern" (disc <-> memoria calculatorului) se face doar cu 1-5 Mo/s. Aceste valori diferă, chiar la acelaşi echipament, în funcţie de modul de lucru selectat:

- sincron 2-5 Mo/s, asincron cu o viteză redusă la jumătate;
- în regim de "salvă", "rafală" 4-5 Mo/s, în regim continuu, susţinut (cînd bufferul SMB nu mai poate acţiona) doar 1-2 Mo/s.

În concluzie, criteriile importante pentru stabilirea performanţelor unei unităţi de disc sînt:

-timpul de poziţionare ;
-timpul de latenţă;
-timpul de transfer al datelor în întregul sistem (dat de interfaţa sau magistrala folosită).

În plus, la unele aplicaţii speciale ce necesită transferul unor volume foarte mari de date la o singură cerere de citire/scriere, trebuie considerat şi timpul de transfer "interior", între disc şi controller.

Există o întreagă gamă de posibilităţi de a spori performanţele unui subsistem de harddisk.

Optimizarea tranzacţiilor

Controller-ul poate optimiza timpul de acces dacă adună toate cererile de scriere şi citire pentru discul respectiv într-un şir de aşteptare (da, şi cererile stau la coadă!), şi lucrează apoi într-o ordine sortată optimal, astfel ca întotdeauna după o cerere să o satisfacă pe cea următoare în sensul distanţei minime pe disc. Deci ordinea nu mai este cea "naturală", în care au fost emise cererile, ci a dispunerii pe pistele suprafeţei de disc folosite în acel moment, a apropierii pe disc.

Un astfel de controller poate aduce, în cazul optim, cînd şirul de aşteptare este suficient de lung, o îmbunătăţire de cca 20-30 % a timpului de acces.

Cazul optimal implică un şir destul de lung de aşteptare a accesului la disc, şir deja format pentru unitatea de harddisk respectivă. Dacă nu-i cazul (de exemplu, dacă nu-s multe cereri, deci şirul de aşteptare este foarte scurt) îmbunătăţirea este insesizabilă.

Pe lingă optimizarea prin sortare după distanţa pe disc, aproape toate controllerele MSCP fac optimizare la cereri de mai multe blocuri, acestea nu vor fi transmise în succesiunea lor naturală, ci în acea ordine care permite harddisk-ului să citească tot în jumătatea interioară a pistei respective, făcînd deci economie de o jumătate de rotaţie. Cum această metodă de optimizare a performanţei este independentă de tipul unităţii de disc, nu este un criteriu de alegere a unui anumit tip de unitate de disc, ci doar o indicaţie (controllerul incorporat să "ştie" acest mod de optimizare).

Pentru a spori performanţa sistemului, se poate "instala" logic un disc mai mare, format din mai multe discuri fizice mai mici. Este inversul partiţionării unui disc în mai multe discuri logice mai mici. Discul logic mare aduce două îmbunătăţiri importante performanţei:

- în primul rînd, unităţile de disc separate se pot poziţiona suprapus;
-apoi, poziţionarea se desfăşoară pe de-a-ntregul atunci cînd capetele de citire/scriere sînt încă pe pista dorită. Probabilitatea respectivă este mai mare la mai multe discuri mici decît la unul mare.

Fragmentarea nu trebuie să fie excesivă - unităţile de disc mici, de 20-40 Mo, au nu numai costul specific mai mare, dar şi timpul de acces mai prost decît un harddisk mai mare, de 120-300 Mo. Se înlocuieşte deci parţial selectarea mecanică a pistelor cu selectarea electronică a unităţilor de disc, iar fiabilitatea sistemului creşte: în caz de pană la un disc, nu sînt afectate toate datele, ci doar cele de pe unitatea respectivă.

Atunci cînd capacitatea maximă de transfer a unui controller sau a unei magistrale de date este depăşită, se poate face o împărţire (repartiţie) a discurilor pe mai multe controllere sau mai multe magistrale de sistem, sau într-un "cluster" cu mai multe calculatoare.

În mod normal, o asemenea împărţire/separare aduce avantaje de viteză, dar mai mici decît se speră.

Cache

O altă posibilitate, ulterioară, de creştere a performanţei harddiskului este folosirea unui controller cu "cache", memorie-tampon de "prindere". ("Cache" se pronunţă "caş" şi vine din franceză: cacher = a ascunde, cache = ascunzătoare).

Mic dicţionar - harddiskuri

Multisegmentare

Unităţile clasice de disc de masă aveau un număr fix de sectoare pe pistă, indiferent de poziţia pistei. Ori, pistele fiind dispuse concentric pe disc, este evident că pe cele exterioare este spaţiu mai mult; cu cît pista este mai aproape de centrul de rotaţie, cu atît mai "strîmt" devine spaţiul pentru sectoarele ce trebuie să încapă pe pistă. Restricţia cu numărul sectoarelor se datora controllerelor - realizate pe atunci cu circuite cu grad de integrare redus (TTL-uri MSI). Nevoia presantă de capacităţi de memorare mai mari - dublată de progresele uriaşe în domeniul circuitelor VLSI, ULSI şi ASIC - a dus la apariţia unor controllere cu funcţii deosebit de complexe, dispunînd de algoritmi sofisticaţi de gestiune a spaţiului fizic al discului. Controllerele cu firmware de multisegmentare împart pistele într-un număr variabil de sectoare, în funcţie de poziţia pistei - mai multe spre exterior, mai puţine spre interior.

Fiecare cerere de scriere pe disc va fi procesată exact ca şi la un controller fără cache: fiecare din blocurile de înregistrat este trimis la capetele de citire/scriere ale unităţii de disc cuplate. În plus însă, controllerul îşi "notează" adresele blocurilor şi conţinutul lor într-o memorie internă, denumită cache. Memoria aceasta are o capacitate variabilă în funcţie de tipul unităţii de harddisk. Astfel, la controllerul QD-35 al firmei Emulex (pentru pînă la patru unităţi de harddisk cu interfaţă SMB, viteza de transfer de max. 3 Mo/s) memoria cache pentru stocarea intermediară a blocurilor de date şi adrese are 1 Mo.

Cînd calculatorul vrea să citească un bloc, controllerul verifică mai întîi dacă blocul respectiv nu se află cumva în memoria cache. Dacă da, va fi transmis imediat calculatorului, fără nici o întîrziere datorită poziţionării, latenţei, etc. Dacă nu se află în cache, va fi citit de pe disc şi transmis, ca în cazul anterior. Pe deasupra, QD-35 reţine şi în această situaţie adresa şi conţinutul blocului.

Cînd memoria-tampon (bufferul-cache) s-a umplut, se va suprascrie, începînd cu blocul care n-a mai fost utilizat/solicitat de cel mai mult timp - o procedură gen FIFO. Se defineşte un coeficient de reuşită, "hit-rate", a "prinderii" datelor în cache. Cu un cache-controller performant, coeficientul de reuşită poate ajunge chiar la 100%: adică, totdeauna cînd calculatorul are nevoie de date de pe disc, ele sînt deja disponibile în cache. Cifra variază în funcţie de împrejurări, de aplicaţia concretă.

Într-adevăr, un controller cu cache este deosebit de recomandabil atunci cînd sînt îndeplinite două sau trei din următoarele condiţii:

-calculatorul are memoria principală relativ redusă;

-unitatea de harddisk primeşte un mare număr de cereri (ca şi la "şirul de aşteptare" - dacă nu sînt multe cereri de citire, nu are ce să optimizeze);

-numărul tipic de blocuri per cerere de citire/scriere este redus;

-se folosesc aplicaţii care, în mod repetat, citesc şi scriu în acelaşi fişier, cu schimbul.

Controllere pentru discuri SMB

Harddisk-urile cu interfaţă SMB sînt performante (capacitate mare, debit ridicat) şi destul de scumpe, ceea ce explică de ce controllerele lor beneficiază de cele mai multe perfecţionări. Emulează protocolul MSCP şi au în consecinţă multe avantaje. Spre deosebire de alte emulări, MSCP lucrează independent de mărimea discului, de structura sa fizică. Softul calculatorului "întreabă" controllerul despre geometria/organizarea harddisk-ului conectat. La MSCP, sistemul vede doar blocuri numerotate în ordine crescătoare, nu piste, cilindrii, sectoare sau feţe de disc - toate aceste detalii fizice sînt "transparente". De asemenea, calculatorul nu mai are treabă cu blocurile cu defecte, bad blocks, tratarea on-line (în timp real) a defectelor fiind şi ea transparentă.

Minimizarea timpului de poziţionare dintre două cereri de citire succesive, menţionată anterior, se obţine cu ajutorul unui aşa numit "algoritm de ascensor". Întodeauna cea mai apropiată cerere din şirul de aşteptare va fi rezolvată - în ordinea distanţei de poziţionare pe disc. Pentru ca o anumită cerere să nu rămînă nerezolvată un timp foarte lung, există un indicator de "tratare cinstită", Fairness Count, şi un şir de aşteptare expres, care asigură soluţionarea rapidă a cererii corespunzătoare.

Controllerele MSCP au şi alte caracteristici avansate, cum ar fi :

-FHS (Fast Head Select). Din moment ce organizarea fizică a discului nu contează, controllerul are posibilitatea de a comuta capetele după necesităţi, pe timpul transferului de date, astfel încît să nu mai fie necesară o rotaţie suplimentară a discului;

- RPS (Rotational Position Sensing), aduce în primul rînd avantajul că, atunci cînd controllerul este legat la mai mult de o unitate de disc, va da întotdeauna cale liberă mai întîi datelor de la/pentru harddisk-ul ale cărui capete au de parcurs drumul minim pînă la poziţionarea corectă, ş. a. m. d.

-A-DMA, accesul direct adaptiv la memorie, reglează eliberarea magistralei de sistem (în general

Noutăţi

Harddisk-urile de 2,5" fiind ultimele apărute, beneficiază de perfecţionări în domeniul algoritmilor specifici precum şi-n hard-ul aferent (circuite VLSI specializate):

- multisegmentare
- cache adaptiv, ce evaluează modul în care datele sînt cerute de sistem şi adoptează automat algoritmul cel mai eficient
- cache cu predicţie, care minimizează timpul de acces la transferuri lungi
- consum redus de energie (2-3 W şi chiar mai puţin în "standby")
- corecţie automată a erorilor (ECC - cod polinomial de 48 biţi, permite corecţia unui sector cu "salvă" de pînă la 11 biţi eronaţi; rata erorilor scade de 1000 de ori (!)
- firmware integrat de autodiagnostic, foarte complet, care identifică şi semnalează prompt condiţiile de eroare
- robusteţe mecanică f. bună - rezistă la şocuri de 100 G cînd nu e în stare de funcţionare - datorită dimensiunilor reduse şi a sistemului automat de "parcare" a capetelor în zone special destinate
- sistem de servopoziţionare hibrid, folosind suprafaţa discului pentru a reduce timpul de acces la valori de ordinul a 10 ms şi servo înglobat (pistă specială cu semnale servo) pentru îmbunătăţirea preciziei de poziţionare a capetelor

Q-bus) cu prioritate mai mică sau mai mare în funcţie de transferul de date dintre harddisk şi calculator.

Fiecare din aceste caracteristici avansate optimizează cîte unul din aspectele performanţei discului : FHS elimină în mare parte timpul de latenţă, RPS îl reduce pe cel de poziţionare, A-DMA diminuează pierderile de timp la transferurile de date.

Funcţia denumită "optimizarea căutării" asigură, împreună cu bufferul de comenzi (ce poate stoca pînă la 13 comenzi succesive), procesarea cererilor de citire/scriere în ordine logică, şi nu pur şi simplu în ordinea apariţiei lor.

În plus controllerul conţine în EPROM (firmware) programe de diagnostic (autotest), de formatare a harddisk-ului şi de tratare a erorilor, astfel că discul poate fi formatat şi verificat fără programe speciale, încărcate de pe dischetă.

Controllerul QD-35 are, în afara funcţiilor descrise mai sus şi a memoriei cache, un procesor intern foarte rapid: un 68020 în capsulă ceramică Quadflatpack. Interesant este cum se face "prinderea" datelor - la eficienţa unei memorii cache contribuind nu numai mărimea acesteia, ci algoritmii utilizaţi. Firma Emulex declară că la proiectarea lui QD-35 a avut în vedere în mod special găsirea unei modalităţi eficiente de "prindere", fără influenţele negative ce se întîlnesc la alte controllere cu cache asupra performanţelor la lucrul cu mai multe unităţi de disc (MDP) sau la sortarea comenzilor (CS).

La QD-35, pentru prima dată, memoria cache este configurabilă după dorinţă. Utilizatorul are posibilitatea de a stabili ce domeniu al discului să fie "prins"şi care nu. În acest scop, la QD-35 (care este un veritabil minisistem) se leagă un terminal, prin intermediul unui cablu plat. Monitorul incorporat lui QD-35 permite setarea şi reglarea parametrilor, supraveghează tranzacţiile cu harddisk-ul şi dă informaţii statistice asupra valorilor timpilor de acces şi a eficacităţii "prinderii", permiţînd astfel stabilirea parametrilor optimi. Atenţie! O memorie cache aşa de mare poate prezenta şi dezavantaje : dacă alimentarea sistemului de calcul nu este asigurată de o sursă neîntreruptibilă (UPS - Uninterruptible Power Supply), în caz de pană de curent toate informaţiile din memoria controllerului se pierd ...

Interfaţa

Evident, controllerul şi unitatea de disc trebuie să aibă aceeaşi interfaţă. Actualmente, majoritatea harddisk-urilor uzuale (de 3,5" şi 5,25", capacităţi pînă în 300 Mo) au controllerul incorporat ("imbedded" - se vorbeşte atunci de interfaţă AT sau IDE - Intelligent Drive Electronics). Sînt mai răspîndite 3 tipuri de interfeţe:

ST506

Interfaţa controllerului original de pe IBM-XT, pentru discuri de capacitate nu prea mare (pînă în 150 Mo). Are două variante, corespunzătoare modului de înregistrare pe disc: MFM (Modified Frequency Modulation) şi RLL (Run Length Limited). Cu RLL, datele sînt procesate suplimentar: prin intermediul unei logici complicate, se obţin coduri în care pot exista între 1 şi 7 zerouri succesive (RLL 1.7) sau între două şi şapte zerouri succesive (RLL 2.7). Se obţine astfel o îmbunătăţire suplimentară a performanţelor, viteza de transfer poate creşte cu 50%. (încă o dată trebuie subliniat că este vorba de transferul între harddisk şi controller. Discul nu trebuie să dea date controllerului cu o viteză mai mare decît cea cu care le poate prelua memoria. Bufferul controllerului nu trebuie să se umple complet într-una sau mai multe rotaţii ale discului. O rată de transfer internă mai ridicată nu are nici un efect dacă viteza de transfer pe magistrala calculatorului este inferioară). Trebuie însă verificat că unitatea de harddisk acceptă şi modul RLL al controllerului: toate unităţile de disc cu ST506 acceptă modul MFM, dar nu toate pot lucra şi în RLL. Cablele sînt identice şi uneori un controller RLL pare să se înţeleagă cu un disc MFM. Dar, în astfel de cazuri, datele Dvs. sînt în mare pericol!

ESDI

Se foloseşte la capacităţi şi viteze mai mari. Viteza maximă de transfer este de 15 Mo/s. Atenţie! Cablurile şi mufele ESDI (Enhanced Small Device Interface) şi ST506 sînt identice ca aspect. Aceasta creează premise de eroare: se poate cupla din greşeală ieşirea unui harddisk ESDI la intrarea unui controller ST506, ceea ce poate duce la defectarea chiar a ambelor componente: se leagă împreună căi cu diferite funcţii.

SCSI

Cel mai nou standard este SCSI (Small Computer System Interface). Spre deosebire de celelalte,

Controllerele - tabel comparativ

	ST506		ESDI	SCSI
	MFM	RLL		asincron/sincron
rata maximală de transfer (interleave 1:1) la -interfaţa serială (Mbit/s)	5	7,5	10	12/40
-unitate (Mbit/s)			10/15	10/15
-sistem (ko/s)	450	650	850/900-1200	850/1200
busul de date	serial	serial	serial	8 (16) biţi
BIOS integrat	nu	da	da	da

nu este o simplă interfaţă de harddisk, ci o veritabilă magistrală de intrare/ieşire. Pe magistrală SCSI se pot lega în paralel mximum 8 surse de date, 8 noduri. Cele 8 noduri pot fi un calculator şi mai multe unităţi de harddisk, sau alte periferice: scanner, unitate de bandă magnetică, imprimantă, disc magneto optic, etc.

Magistrala SCSI "dă" mai multă inteligenţă perifericelor şi cere mai puţin plăcii de legătură a calculatorului. Discurile sub SCSI pot lucra în paralel, dacă sistemul de operare acceptă mai mult de o singură cerere de harddisk la un moment dat, adică numai cu sisteme de operare multitasking precum Unix, OS/2 sau Novell, nu MS-DOS. Şi cu asta se trece şi la capitolul următor:

Sistemul de operare

Iniţial, am neglijat timpul de overhead al sistemului de operare, fiind acelaşi pentru toate tipurile de harddisk-uri. Totuşi, sistemul de operare afectează modul de lucru cu discul, este un criteriu important de decizie în alegerea tipului de harddisk. Astfel, putem avea un harddisk de capacitate foarte mare, dar sistemul de operare să nu se potrivească. De exemplu, MS-DOS "consideră" că discurile au maximum 1024 de cilindri. Această limitare a sistemului de operare poate fi depăşită chiar dacă harddisk-ul are mai mult de o mie de cilindri (piste per faţă de disc): cineva, de regulă controllerul, va masca realitatea de pe disc, disponibilizînd însă mai multe feţe de disc (false) sau mai multe sectoare pe o pistă (false). În această tehnică lucrează controllerele din seria ACB23 a lui Adaptec sau WD1007Val lui Western Digital.

MS-DOS este un sistem de operare single-task; dar sub VMS, OS, Novell şi Unix se poate lucra în multitasking, cu mai multe unităţi de disc mici - cum s-a arătat mai înainte, pînă ce unul îşi poziţionează capetele, ne adresăm altuia pentru transferul de date.

Transportabilitatea

Este un criteriu de selecţie esenţial atunci cînd se impune ca datele să poată fi transportate rapid şi uşor de la un calculator la altul, sau cînd sînt date sensibile, ce trebuiesc protejate.

La harddisk-urile amovibile, unitatea este într-o cutie-sertar cu conectori, cutie care poate fi uşor detaşată pentru transport. Unitatea de bază este un şasiu, care conţine sursa de alimentare şi conectorii, şi poate accepta unul sau două containere cu discuri. Containerele, etanşe, cuprind un mediu controlat, clasă 100 (sub 100 de particule de praf / ft^3), în care funcţionează discurile.

Cum fiecare container cuprinde unitatea de harddisk completă, costul este ridicat, atît pentru subsistemul propriu-zis, cît şi pentru suportul de date. O soluţie mult mai economică o oferă discurile-cartuş (cartridge), gen Bernoulli-Box, la care se schimbă numai suportul de date, nu toată unitatea.

Totuşi, harddisk-urile amovibile în container se utilizează pe larg, avînd două avantaje majore:

-sînt cele mai bune atunci cînd este necesar să se transporte cantităţi mari de date, dar nu există timpul pentru a le capta pe bandă sau casetă magnetică;

-sînt foarte bune şi atunci cînd datele unui sistem trebuiesc puse în siguranţă, în întregime şi în mod rapid - de exemplu containerele se încuie peste noapte în seif.

Faţă de transportul sau asigurarea (back-up) datelor prin copierea pe bandă, harddisk-urile amovile prezintă un timp de acces mult mai redus, iar probabilitatea de apariţie a unor erori la copiere este considerabil diminuată. Un alt suport de date interschimbabil ce le-ar putea face concurenţă este discul optic reînregistrabil (disc magneto-optic), dar acesta prezintă actualmente inconveniente ca:

-deşi suportul de date este ieftin pentru capacitatea sa, unitatea de disc magneto-optică este foarte scumpă;

-timpii de acces sînt mari, de 4 ori mai mari la citire şi de 10 ori la scriere, în timp ce discurile amovibile au timpi de răspuns comparabili cu cei ai harddisk-urilor Winchester normale;

-şi rata de transfer este mai scăzută, 0,5-1 Mo/s faţă de 2-5 Mo/s la harddisk-urile normale.

Alte criterii

Actualmente, cele mai răspîndite sînt harddisk-urile de 5,25" (fi 13 cm), care există în două înălţimi diferite ale cutiei:

-înălţime întreagă (FH - full height, un model de 10,5 cm), cu capacităţi de peste 300 Mo, ajungînd pînă la 1200 Mo;

-semi-înălţime (HH - half height, 5,25 cm), relativ răspîndită la discurile pînă în 200 Mo.

Discurile de 8" (20 cm), cu interfeţe tip SMB sau SCSI, au capacităţi de pînă la 2000-3200 Mo.

Cele mai mari unităţi de disc au diametrul de 9", dar sînt puţin răspîndite, nefiind competitive nici ca performanţe, nici ca preţ (costisitoare, dimensiuni, consum şi greutate foarte ridicate).

ing. Cristian Malide, I.C.C.E.

Filecard

Este numele sub care sînt cunoscute harddiskurile ce se montează direct într-un slot liber al PC-ului, fiind solidare cu cartela care se introduce şi pe care sînt montate. De regulă sînt mai scumpe decît harddisk-urile clasice; sînt o alternativă bună cînd trebuie extinsă capacitatea disc dar controllerul este deja "epuizat" sau nu mai este loc prevăzut pentru o unitate suplimentară de harddisk.

Firma	Model	Preţ (DM)	Formă	Capacitate (Mo)	Timp de acces (ms)	Rata de transfer (Mo/s)	Tip interfaţă	Mod de înregistrare	Cache integrat (ko)	Cilindri/capete	MTBF (mii ore)
Adcomp	AL D Turbo-Disk		ext. 2 x 5,25" FH	/330	10,5	5	SCSI	Zoom-bit recording	32		
Conner	C P-3044	581	3,5"-Winchester	43/40	25	1,5	AT	RLL (2,7)			
	CP-3104	1026	-||-	/105	25	1,25	AT	RLL (2,7)			
	CP-3204F	1676	-||-	/212	16	1,5	AT	RLL (2,7)			
	HOPI CP30104		3,5"	/120	16	1,5/4,0	AT/EISA	RLL	64		
	RAMBO CP3204F		3,5"	/212	16	1,5/5,0	AT	RLL	64		
Data General	6557	2428	3,5"	/104	25	1,25	IDE	RLL			
Daimler International	Legacy Mass Tower I/380	9151	subsistem extern	/380	>15	2,7	SCSI	RLL			
Fujitsu	M2611SA		3,5" HH	/45	25	1,5	SCSI	RLL (1,7)	24	1334/2	50
	M2614ESA	2.438	3,5"	/180	20	1,25	SCSI	RLL (1,7)	24	1334/4	
	M2266SA	9776	5,25"	/1079	14,5	4	SCSI	RLL (1,7)	128	1658/15+1	200
	M2263SA	7085	5,25"	/672	16	4	SCSI	RLL (1,7)	64	1658/15+1	200
	M2246SA	1.850		172/141	25		SCSI				
	M2622SA	2.630	3,5" FH	/330	12,5	3	SCSI-2	RLL (1,7)	4 x 60	1435/7+1	75
	M2623SA	3.050	3,5" FH	/425	12	3	SCSI-2	RLL (1,7)	4 x 60	1435/9+1	
	M2624SA	3.480	3,5" FH	/520	12	3	SCSI-2	RLL (1,7)	4 x 60	1435/11+1	
Hewlett-Packard	HP C2233S		3,5"	/234	13	2,5	SCSI-2		64	2172/15	
Hitachi	DK516C-16		5,25"	1652/	13,5		SCSI				
IBM	Modell 60/80	1.250	5,25"	/44	40		ST-506				
	PS/2	2.880	3,5"	/60	27		ESDI				
	PS/2	2095	3,5"	/60	23		SCSI				
	PS/2	12.860	5,25"	/314	23		ESDI				
Kyocera	KC40 GA		3,5"	/40	28		ST506	RLL		1069/2	
Maxtor	EM 8051A3		3,5"	/42	28	1,0	IDE-AT	RLL	32		
	XT-2190		5,25"	/160	29	5	ST506	MFM	-		
	XT-4380		5,25"	/338	16	4,8	SCSI/ESDI	RLL (2,7)	45/SCSI		
Micropolis	1528-15		5,25"	/1354	14	1,6/5,0	SCSI-2	RLL	256	2106/15	
Mitsubishi Electric	MR 335		3,5"	/40	20	5	ST506	MFM	-	612/5	
NCR		1.960	3,5"	/42	25	1,25	AT	RLL	-		
NEC	D3761		3,5"	/733	20		AT	RLL (2,7)		733/8	
Plus Development	Filecard80	1.664	AT 3,5" Winchester	/80	28	0,7	EISA	RLL	-		
Procom Technology	PhD 45	1.450	3,5"	/44,7	25	5,0	IBM-PS/2	MFM	-		
	Pira 50	1.505	3,5" Winchester	/45	25	0,625	ST-506	MFM			
	Si 200	3.595	3,5"	/209	18	12	Compaq Deskpro	RLL/SCSI			
Quantum	LPS 52S	718	3,5"	/52	<17	2	SCSI		64	1219/2	
	LPS 52 AT	798	3,5"	/52	<17	4	AT	RLL (1,7)	64	1219/2	
	LPS 105 AT	1.398	3,5"	/105	<17	0,8	IDE	RLL (1,7)	64	1219/2	
	LPS 105 S	1.173	3,5"	/105	<17	2/4	SCSI		64	1219/2	
	Pro Drive 40S	618	3,5"	/42	19	2/4	SCSI	RLL (2,7)		834/3	
	Pro Drive 80S	927	3,5"	/84	19	2/4	SCSI	RLL (2,7)		834/6	
	Pro Drive 120 AT		3,5"	/120	<15	4	AT	RLL (1,7)		1123/5	
	Pro Drive 210AT		3,5"	/210	<15	4	AT	RLL (1,7)		1156/7	
Rodime	RO 3000 AT		3,5"	45-210	18	max 2,5	AT (IDE)	RLL (2,7)	32/64		
Seagate	ST 325N		3,5"	/21,4	28		SCSI	RLL (2,7)			
	ST125N		3,5"	/21,5	28	1,5	SCSI	RLL (2,7)			
	ST138N		3,5"	/32,2		1,5	SCSI	RLL (2,7)			
	ST157N		3,5"	/48,6	20	1,5	SCSI	RLL (2,7)			
	ST177N		3,5"	/60,8	20	1,5	SCSI	RLL (2,7)			
	ST1096N		3,5"	/83,9	15	1,5	SCSI	RLL (2,7)			
	ST1126N		3,5"	/107,0	15	1,5	SCSI	RLL (2,7)			
	ST1133N		3,5"	/113,4	15	1,25	SCSI	RLL (2,7)			
	ST1162N		3,5"	/137,5	15	5	SCSI	RLL (2,7)			
	ST1186N		3,5"	/158,8	15	5	SCSI	RLL (2,7)			
	ST1201N		3,5"	/171,9	15	5	SCSI	RLL (2,7)			
	ST1239N		3,5"	/204,2	15	5	SCSI	RLL (2,7)			
Toshiba	MK 134FA		3,5"	/40	25	5	ST-506				
Western Digital	Filecard 40	752	3,5"	/40	29	0,9	ISA	RLL			

Rutine care clasifică

Aceste rutine clasifică caracterele ASCII în litere, caractere de control, semne de punctuaţie, majuscule, etc.

```
        isalnum(c)                                          <ctype.h>
        isalpha(c)                                          <ctype.h>
        isascii(c)                                          <ctype.h>
        iscntrl(c)                                          <ctype.h>
        isdigit(c)                                          <ctype.h>
        isgraph(c)                                          <ctype.h>
        islower(c)                                          <ctype.h>
        isprint(c)                                          <ctype.h>
        ispunct(c)                                          <ctype.h>
        isspace(c)                                          <ctype.h>
        isupper(c)                                          <ctype.h>
        isxdigit(c)                                         <ctype.h>
```

Rutine de conversie

Aceste rutine convertesc şiruri alfanumerice în diferite reprezentări numerice şi invers, din majuscule în minuscule şi invers

```
double      atof(const char *s)                             <stdlib.h>
int         atoi(const char *s)                             <stdlib.h>
long        atol(const char *s)                             <stdlib.h>
char        *ecvt(double value, int ndig, int *dec, int *sign)   <stdlib.h>
char        *fcvt(double value, int ndig, int *dec, int *sign)   <stdlib.h>
char        *gcvt(double value, int ndec, char *buf)        <stdlib.h>
char        itoa(int value, char *string, int radix)        <stdlib.h>
char        ltoa(long value, char *string, int radix)       <stdlib.h>
double      strtod(const char *s, char **endptr)            <stdlib.h>
long        strtol(const char *s, char **endptr, int radix) <stdlib.h>
unsigned long strtoul(const char *s, char **endptr, int radix)   <stdlib.h>
```

```
            toascii(c)                                      <ctype.h>
            _tolower(c)                                     <ctype.h>
int         tolower(int ch)                                 <ctype.h>
            _toupper(c)                                     <ctype.h>
int         toupper(int ch)                                 <ctype.h>
char        ultoa(unsigned long value, char *string, int radix)   <stdlib.h>
```

Rutine de control al directoarelor

Aceste rutine manipulează directoare şi nume de căi

```
int         chdir(const char *path)                         <dir.h>
int         findfirst(const char *pathname, struct ffblk *ffblk,
                int attrib)                                 <dir.h>
int         findnext(struct ffblk *ffblk)                   <dir.h>
void        fnmerge(char *path, const char *drive, const char *dir,
                const char *name, const char *ext)          <dir.h>
int         fnsplit(char *path, const char *drive, const char *dir,
                const char *name, const char *ext)          <dir.h>
int         getcurdir(int drive, char *directory)           <dir.h>
char        *getcwd(char *buf, int buflen)                  <dir.h>
int         getdisk(void)                                   <dir.h>
int         mkdir(const char *path)                         <dir.h>
char        *mktemp(char *template)                         <dir.h>
int         rmdir(const char *path)                         <dir.h>
char        *searchpath(const char *file)                   <dir.h>
int         setdisk(int drive)                              <dir.h>
```

Rutine de diagnosticare

Aceste rutine asigură capacitatea de depanare înglobată în programe

```
void        assert(int test)                                <assert.h>
int         matherr(struct  exception *e)                   <math.h>
```


Turbo C++ 1.0
Memento - Funcţii

```
int         freemem(unsigned segx)                          <dos.h>
void        geninterrupt(int intr_num)                      <dos.h>
int         getcbrk(void)                                   <dos.h>
void        getdfree(unsigned char drive, struct dfree *dtable)   <dos.h>
char far    *getdta(void)                                   <dos.h>
void        getfat(unsigned char drive, struct fatinfo *dtable)   <dos.h>
void        getfatd(struct fatinfo *dtable)                 <dos.h>
unsigned    getpsp(void)                                    <dos.h>
void        interrupt (*getvect(int interrupno)) ()         <dos.h>
int         getverify(void)                                 <dos.h>
void        harderr(int (*handler) ())                      <dos.h>
void        hardresume(int axret)                           <dos.h>
void        hardretn(int retn)                              <dos.h>
int         inport(int portid)                              <dos.h>
unsigned char  inportb(int portid)                          <dos.h>
int         int86(int intno, union REGS *inregs, union REGS *outregs)   <dos.h>
int         int86x(int intno, union REGS *inregs, union REGS *outregs,
                struct SREGS *segregs)                      <dos.h>
int         intdos(union REGS *inregs, union REGS *outregs)     <dos.h>
int         intdosx(union REGS *inregs, union REGS *outregs,
                struct SREGS *segregs)                      <dos.h>
void        intr(int intno, struct REGPACK *preg)           <dos.h>
void        keep(unsigned char status, unsigned size)       <dos.h>
void far *MK_FP(unsigned seg, insigned ofs)                 <dos.h>
void        outport(int portid, int value)                  <dos.h>
void        outportb(int portid, unsigned char value)       <dos.h>
char        *parsfnm(const char *cmdline, struct fcb *fcb, int opt)   <dos.h>
int         peek(unsigned segment, unsigned offset)         <dos.h>
char        peekb(unsigned segment, unsigned offset)        <dos.h>
void        poke(unsigned segment, unsigned offset, int value)   <dos.h>
void        pokeb(unsigned segment, unsigned offset, char value)   <dos.h>
int         randbrd(struct fcb *fcb, int rcnt)              <dos.h>
int         randbwr(struct fcb *fcb, int rcnt)              <dos.h>
void        segread(struct SREGS *segp)                     <dos.h>
int         setcbrk(int cbrkvalue)                          <dos.h>
void        setdta(char far *dta)                           <dos.h>
void        setvect(int interruptno, void interrupt (*isr) ( ))   <dos.h>
void        setverify(int value)                            <dos.h>
```

```
void        sleep(unsigned seconds)                         <dos.h>
int         unlink(const char *filename)                    <dos.h, io.h, stdio.h>
```

Rutine de manipulare şiruri şi zone de memorie

```
void        *memccpy(void *dest, const void *src, int c,
                size_t n)                                   <mem.h, string.h>
void        memchr(const void *s, int c, size_t n)          <mem.h, string.h>
int         memcmp(const void *s1, const void *s2, size_t n)    <mem.h, string.h>
void        *memcpy(void *dest, const void *src, size_t n)  <mem.h, string.h>
void        *memmove(void *dest, const void *src, size_t n) <mem.h, string.h>
void        *memset(void *s, int c, size_t n)               <mem.h, string.h>
void        movedata(unsigned srcseg, unsigned srcoff, unsigned destseg,
                unsigned destoff, size_t n)                 <mem.h, string.h>
void        movmem(void *src, void *dest, unsigned length)  <mem.h, string.h>
void        setmem(void *dest, int length, char value)      <mem.h>
char        *stpcpy(char *dest, const char *src)            <string.h>
char        *strcat(char *dest, const char *src)            <string.h>
char        *strchr(const char *s, int c)                   <string.h>
int         strcmp(const char *s1, const char *s2)          <string.h>
int         strcoll(char *s1, char *s2)                     <string.h>
char        *strcpy(char *dest, const char *src)            <string.h>
size_t      strcspn(const char *s1, const char *s2)         <string.h>
char        *strdup(const char *s)                          <string.h>
char        *strerror(int errnum)                           <string.h>
size_t      strlen(const char *s)                           <string.h>
char        *strlwr(char *s)                                <string.h>
char  *strncat(char *dest, const char *src, size_t maxlen)  <string.h>
int         strncmp(const char *s1, const char *s2, size_t maxlen)   <string.h>
char        *strncpy(char *dest, const char *src, size_t maxlen)   <string.h>
char        *strnset(int *s, int ch, size_t n)              <string.h>
char        *strpbrk(const char *s1, const char *s2)        <string.h>
char        *strrchr(const char *s, int c)                  <string.h>
char        *strrev(char *s)                                <string.h>
char        *strset(char *s, int ch)                        <string.h>
size_t      strspn(const char *s1, const char *s2)          <string.h>
char        *strstr(const char *s1, const char *s2)         <string.h>
char        *strtok(char *s1, const char *s2)               <string.h>
```

Turbo C++ 1.0
Memento - funcţii

```
void        perror(const char *s)                                   <errno.h>
```

Rutine grafice

Aceste rutine permit realizarea de grafică şi text pe ecran

```
void far    arc(int x, int y, int stangle, int endangle, int radius)          <graphics.h>
void far    bar(int left, int top, int right, int bottom)                     <graphics.h>
void far    bar3d(int left, int top, int right, int bottom,
                  int depth, int topflag)                                     <graphics.h>
void far    circle(int x, int y, int radius)                                  <graphics.h>
void far    cleardevice(void)                                                 <graphics.h>
void far    clearviewport(void)                                               <graphics.h>
void far    closegraph(void)                                                  <graphics.h>
void far    detectgraph(int far *graphdriver, int far *graphmode)             <graphics.h>
void far    drawpoly(int numpoints, int far *polypoints)                      <graphics.h>
void far    ellipse(int x, int y, int stangle, int endangle,
                  int xradius, int yradius)                                   <graphics.h>
void far    fillellipse(int x, int y, int xradius, int yradius)               <graphics.h>
void far    fillpoly(int numpoints, int far *polypoints[])                    <graphics.h>
void far    floodfill(int x, int y, int border)                               <graphics.h>
void far    getarccoords(struct arccoordstype, far *arccords)                 <graphics.h>
void far    getaspectratio(int far *xasp, int far *yasp)                      <graphics.h>
int far     getbkcolor(void)                                                  <graphics.h>
int far     getcolor(void)                                                    <graphics.h>
struct palettetype *far  getdefaultpalette(void)                              <graphics.h>
char *far   getdrivername(void)                                               <graphics.h>
void far    getfillpattern(char far *pattern)                                 <graphics.h>
void far    getfillsettings(struct fillsettingstype far *fillinfo)            <graphics.h>
int far     getgraphmode(void)                                                <graphics.h>
void far    getimage(int left, int top, int right, int bottom,
                  void far *bitmap)                                           <graphics.h>
void far    getlinesettings(struct linesettingstype far *lineinfo)            <graphics.h>
int far     getmaxcolor(void)                                                 <graphics.h>
int far     getmaxmode(void)                                                  <graphics.h>
int far     getmaxx(void)                                                     <graphics.h>
int far     getmaxy(void)                                                     <graphics.h>
char * far  getmodename(int mode_number)                                      <graphics.h>
void far    getmoderange(int graphdriver, int far *lomode,
                  int far *himode)                                            <graphics.h>
void far    getpalette(struct palettetype far *palette)                       <graphics.h>
int far     getpalettesize(void)                                              <graphics.h>
unsigned far  getpixel(int x, int y)                                          <graphics.h>
void far    gettextsettings(struct textsettingstype far *
                  texttypeinfo)                                               <graphics.h>
void far    getviewsettings(struct viewporttype far *viewport)                <graphics.h>
int far     getx(void)                                                        <graphics.h>
int far     gety(void)                                                        <graphics.h>
void far    graphdefaults(void)                                               <graphics.h>
char *far   grapherrormsg(int errorcode)                                      <graphics.h>
void far    _graphfreemem(void far *ptr, unsigned size)                       <graphics.h>
void far * far  _graphgetmem(unsigned size)                                   <graphics.h>
int far     graphresult(void)                                                 <graphics.h>
unsigned far  imagesize(int left, int top, int right, int bottom)             <graphics.h>
void far    initgraph(int far *graphdriver, int far *graphmode,
                  char far *pathtodriver)                                     <graphics.h>
int far     installuserdriver(char far *name,
                  int huge (*detect) (void))                                  <graphics.h>
int far     installuserfont(char far *name)                                   <graphics.h>
void far    line(int x1, int y1, int x2, int y2)                              <graphics.h>
void far    linerel(int dx, int dy)                                           <graphics.h>
void far    lineto(int x, int y)                                              <graphics.h>
void far    moverel(int dx, int dy)                                           <graphics.h>
void far    moveto(int x, int y)                                              <graphics.h>
void far    outtext(char far *textstring)                                     <graphics.h>
void far    outtextxy(int x, int y, char far *textstring)                     <graphics.h>
void far    pieslice(int x, int y, int stangle, int endangle, int radius)     <graphics.h>
void far    putimage(int left, int top, void far *bitmap, int op)             <graphics.h>
void far    putpixel(int x, int y, int color)                                 <graphics.h>
void far    rectangle(int left, int top, int right, int bottom)               <graphics.h>
int         registerbgidriver(void (*driver) (void))                          <graphics.h>
int         registerbgifont(void (*font)(void))                               <graphics.h>
void far    restorecrtmode(void)                                              <graphics.h>
```


Turbo C++ 1.0
Memento - funcţii


```
char        *strupr(char *s)                                        <string.h>
size_t      strxfrm(char *s1, char *s2, size_t n)                   <string.h>
```

Rutine matematice

```
[int | double]  abs([int | complex] x)                   <math.h, complex.h, stdlib.h>
[double | complex]  acos([double | complex] x)                  <complex.h, math.h>
double      arg(complex z)                                          <complex.h>
[double | complex]  asin([double | complex] x)                  <complex.h, math.h>
[double | complex]  atan([double | complex] x)                  <complex.h, math.h>
double      atan2(double y, double x)                           <complex.h, math.h>
double      atof(const char *s)                                  <stdlib.h, math.h>
int         atoi(const char *s)                                     <stdlib.h>
long        atol(const char *s)                                     <stdlib.h>
bcd         bcd(int x)                                               <bcd.h>
bcd         bcd(double x)                                            <bcd.h>
bcd         bcd(double x, int decimals)                              <bcd.h>
double      cabs(struct complex z)                                  <math.h>
double      ceil(double x)                                          <math.h>
unsigned int  _clear87(void)                                        <float.h>
complex     complex(double real, double imag)                       <complex.h>
double      conj(complex z)                                         <complex.h>
unsigned int  _control87(unsigned int new, unsigned int mask)       <float.h>
[double | complex]  cos([double | complex] z)           <complex.h, math.h>
[double | complex]  cosh([double | complex] z)          <complex.h, math.h>
div_t       div(int numer, int denom)                               <math.h>
char        *ecvt(double value, int ndig, int *dec, int *sign)      <stdlib.h>
[double | complex]  exp([double | complex] z)                   <math.h>
double      fabs(double x)                                          <math.h>
char        *fcvt(double value, int ndig, int *dec, int *sign)      <stdlib.h>
double      floor(double x)                                         <math.h>
double      fmod(double x, double y)                                <math.h>
void        _fpreset(void)                                          <float.h>
double      frexp(double x, int *exponent)                          <math.h>
char        *gcvt(double value, int ndec, char *buf)                <stdlib.h>
double      hypot(double x, double y)                               <math.h>
double      imag(complex z)                                         <complex.h>
char        *ltoa(long value, char *string, int radix)              <stdlib.h>
long int    labs(long int x)                                        <stdlib.h>
double      ldexp(double x, int exp)                                <math.h>
ldiv_t      ldiv(long int numer, long int denom)                    <math.h>
[double | complex]  log([double | complex] x)           <complex.h, math.h>
double      log10(double x)                             <complex.h, math.h>
unsigned long  _lrotl(unsigned long val, int count)                 <stdlib.h>
unsigned long  _lrotr(unsigned long val, int count)                 <stdlib.h>
            _mexcep(enumerated type)                                <math.h>
int         matherr(struct exception *e)                            <math.h>
double      modf(double x, double *ipart)                           <math.h>
double      norm(complex x)                                         <complex.h>
complex     polar(double mag, double angle)                         <complex.h>
double      poly(double x, int degree, double coeffs [])            <math.h>
[double | complex]  pow([double | complex] x,
                  [double | complex] y)                     <complex.h, math.h>
double      pow10(int p)                                            <math.h>
int         rand(void)                                              <stdlib.h>
int         random(int num)                                         <stdlib.h>
void        randomize(void)                                         <stdlib.h>
double      real(complex x)                                         <complex.h>
unsigned    _rotl(unsigned val, int count)                          <stdlib.h>
unsigned    _rotr(unsigned val, int count)                          <stdlib.h>
[double | complex]  sin([double | complex] z)           <complex.h, math.h>
[double | complex]  sinh([double | complex] z)          <complex.h, math.h>
[double | complex]  sqrt([double | complex] x)          <complex.h, math.h>
void        srand(unsigned seed)                                    <stdlib.h>
unsigned int  _status87(void)                                       <float.h>
double      strtod(const char *s, char **endptr)                    <stdlib.h>
long        strtol(const char *s, char **endptr, int radix)         <stdlib.h>
unsigned long  strtoul(const char *s, char **endptr, int radix)     <stdlib.h>
[double | complex]  tan([double | complex] x)           <complex.h, math.h>
[double | complex]  tanh([double | complex] x)          <complex.h, math.h>
char        *ultoa(unsigned long value, char *string, int radix)    <stdlib.h>
```

Rutine de gestiune a memoriei

```
int         allocmem(unsigned size, unsigned *segp)                 <dos.h>
int         brk(void *addr)                                         <alloc.h>
```

void far	sector(int x, int y, int stangle, int endangle, int xradius, int yradius)	<graphics.h>
void far	setactivepage(int page)	<graphics.h>
void far	setallpalette(struct palettetype far *palette)	<graphics.h>
void far	setaspectratio(int xasp, int yasp)	<graphics.h>
void far	setbkcolor(int color)	<graphics.h>
void far	setcolor(int color)	<graphics.h>
void far	setfillpattern(char far *upattern, int color)	<graphics.h>
void far	setfillstyle(int pattern, int color)	<graphics.h>
unsigned far	setgraphbufsize(unsigned bufsize)	<graphics.h>
void far	setgraphmode(int mode)	<graphics.h>
void far	setlinestyle(int linestyle, unsigned upattern, int thickness)	<graphics.h>
void far	setpalette(int colornum, int color)	<graphics.h>
void far	setrgbpalette(int colornum, int red, int green, int blue)	<graphics.h>
void far	settextjustify(int horiz, int vert)	<graphics.h>
void far	settextstyle(int font, int direction, int charsize)	<graphics.h>
void far	setusercharsize(int multx, int divx, int multy, int divy)	<graphics.h>
void far	setviewport(int left, int top, int right, int bottom, int clip)	<graphics.h>
void far	setvisualpage(int page)	<graphics.h>
void far	setwritemode(int mode)	<graphics.h>
int far	textheight(char far *textstring)	<graphics.h>
int far	textwidth(char far *textstring)	<graphics.h>

Rutine de intrare/ieşire

Aceste rutine asigură posibilităţi de intrare /ieşire la nivel stream şi DOS

int	access(const char *filename, int amode)	<io.h>
char	*cgets(char *str)	<conio.h>
int	_chmod(const char *path, int func [, int attrib])	<io.h>
int	chmod(const char *path, int amode)	<io.h>
int	chsize(int handle, long size)	<io.h>
void	clearerr(FILE *stream)	<stdio.h>
int	_close(int handle)	<io.h>
int	close(int handle)	<io.h>
int	cprintf(const char *format [, argument,...])	<conio.h>
int	cputs(const char *str)	<conio.h>
int	_creat(const char *path, int attrib)	<io.h>
int	creat(const char *path, int amode)	<io.h>
int	creatnew(const char *path, int mode)	<io.h>
int	creattemp(char *path, int attrib)	<io.h>
int	cscanf(char *format [, address, ...])	<conio.h>
int	dup(int handle)	<io.h>
int	dup2(int oldhandle, int newhandle)	<io.h>
int	eof(int handle)	<io.h>
int	fclose(FILE *stream)	<stdio.h>
int	fcloseall(void)	<stdio.h>
FILE	*fdopen(int handle, char *type)	<stdio.h>
int	feof(FILE *stream)	<stdio.h>
int	ferror(FILE *stream)	<stdio.h>
int	fflush(FILE *stream)	<stdio.h>
int	fgetc(FILE *stream)	<stdio.h>
int	fgetchar(void)	<stdio.h>
int	fgetpos(FILE *stream, fpos_t *pos)	<stdio.h>
char	*fgets(char *s, int n, FILE *stream)	<stdio.h>
long	filelength(int handle)	<io.h>
int	fileno(FILE *stream)	<stdio.h>
int	flushall(void)	<stdio.h>
FILE	*fopen(const char *filename, const char *mode)	<stdio.h>
int	fprintf(FILE *stream, const char *format [, argument, ...])	<stdio.h>
int	fputc(int c, FILE *stream)	<stdio.h>
int	fputchar(int c)	<stdio.h>
int	fputs(const char *s, FILE *stream)	<stdio.h>
size_t	fread(void *ptr, size_t size, size_t n, FILE *stream)	<stdio.h>
FILE	*freopen(const char *filename, const char *mode, FILE *stream)	<stdio.h>
int	fscanf(FILE *stream, const char *format [, address, ...])	<stdio.h>
int	fseek(FILE *stream, long offset, int whence)	<stdio.h>
int	fsetpos(FILE *stream, const fpos_t *pos)	<stdio.h>
int	fstat(int handle, struct stat *statbuf)	<sys\stat.h>
long	ftell(FILE *stream)	<stdio.h>
size_t	fwrite(const void *ptr, size_t size, size_t n, FILE *stream)	<stdio.h>

Turbo C++ 1.0
Memento - Funcţii

void	*calloc(size_t nitems, size_t size)	<alloc.h>
unsigned [long]	coreleft(void)	<alloc.h>
void far	*farcalloc(unsigned long nunits, unsigned long unitsz)	<alloc.h>
unsigned long	farcoreleft(void)	<alloc.h>
void	farfree(void far *block)	<alloc.h>
int	farheapcheck(void)	<alloc.h>
int	farheapcheckfree(unsigned int fillvalue)	<alloc.h>
int	farheapchecknode(void *node)	<alloc.h>
int	farheapfillfree(unsigned int fillvalue)	<alloc.h>
int	farheapwalk(struct farheapinfo *hi)	<alloc.h>
void far	*farmalloc(unsigned long nbytes)	<alloc.h>
void far	*farrealloc(void far *oldblock, unsigned long nbytes)	<alloc.h>
void	free(void *block)	<alloc.h, stdlib.h>
int	heapcheck(void)	<alloc.h>
int	heapcheckfree(unsigned int fillvalue)	<alloc.h>
int	heapchecknode(void *node)	<alloc.h>
int	heapwalk(struct heapinfo *hi)	<alloc.h>
void	*malloc(size_t size)	<alloc.h, stdlib.h>
void	*realloc(void *block, size_t size)	<alloc.h, stdlib.h>
void	*sbrk(int incr)	<alloc.h>
int	setblock(unsigned segx, unsigned newsize)	<dos.h>

Alte rutine

void	delay(unsigned milliseconds)	<dos.h>
struct lconv	*localeconv(void)	<locale.h>
void	longjmp(jmp_buf jmpb, int retval)	<setjmp.h>
void	nosound(void)	<dos.h>
int	setjmp(jmp_buf jmpb)	<setjmp.h>
char	*setlocale(int category, char *locale)	<locale.h>
void	sound(unsigned frequency)	<dos.h>

Rutine de control procese

Aceste rutine pornesc şi opresc procese noi dinăuntrul altor procese

void	abort(void)	<stdlib.h, process.h>
int	execl(char *path, char *arg0, .., NUll)	<process.h>
int	execle(char *path, char *arg0, .., NUll, char **env)	<process.h>
int	execlp(char *path, char *arg0, ..)	<process.h>
int	execlpe(char *path, char *arg0, .., NUll, char **env)	<process.h>
int	execv(char *path, char *argv[])	<process.h>
int	execve(char *path, char *argv[],char **env)	<process.h>
int	execvp(char *path, char *argv[])	<process.h>
int	execvpe(char *path, char *argv[], char **env)	<process.h>
void	_exit(int status)	<process.h, stdlib.h>
void	exit(int status)	<process.h, stdlib.h>
	getpid	<process.h>
int	raise(int sig)	<signal.h>
void	(*signal(int sig, void (*func) int sig[, int subcode]))(int)	<signal.h>
int	spawnl(int mode, char *path, char *arg0, .., NUll)	<process.h>
int	spawnle(int mode, char *path, char *arg0, .., NUll, char *envp[])	<process.h>
int	spawnlp(int mode, char *path, char *arg0, .., NUll)	<process.h>
int	spawnlpe(int mode, char *path, char *arg0, .., NUll, char *envp[])	<process.h>
int	spawnv(int mode, char *path, char *argv[])	<process.h>
int	spawnve(int mode, char *path, char *argv[], char *envp[])	<process.h>
int	spawnvp(int mode, char *path, char *argv[])	<process.h>
int	spawnvpe(int mode, char *path, char *argv[], char *envp[])	<process.h>

Rutine standard

void	abort(void)	<stdlib.h>
int	abs(int x)	<math.h, stdlib.h>
double	abs(complex x)	<complex.h>
int	atexit(atexit_t func)	<stdlib.h>
double	atof(const char *s)	<math.h, stdlib.h>
int	atoi(const char *s)	<stdlib.h>
long	atol(const char *s)	<stdlib.h>
void	*bsearch(const void *key, const void *base, size_t *nelem, size_t width, int (*fcmp)(const void*, const void*))	<stdlib.h>
void	*calloc(size_t nitems, size_t size)	<alloc.h, stdlib.h>

Turbo C++ 1.0
Memento - funcţii

```
int     getc(FILE *stream)                                          <stdio.h>
int     getch(void)                                                 <conio.h>
int     getchar(void)                                               <stdio.h>
int     getche(void)                                                <conio.h>
int     getftime(int handle, struct ftime *ftimep)                  <io.h>
char    *getpass(const char *prompt)                                <conio.h>
char    *gets(char *string)                                         <stdio.h>
int     getw(FILE *stream)                                          <stdio.h>
int     ioctl(int handle, int func [, void *argdx, int argcx ])     <io.h>
int     isatty(int handle)                                          <io.h>
int     kbhit(void)                                                 <conio.h>
int     lock(int handle, long offset, long length)                  <io.h>
long    lseek(int handle, long offset, int fromwhere)               <io.h>
int     _open(const char *filename, int oflags)                     <io.h>
int     open(const char *path, int access [ , unsigned mode ])      <io.h>
void    perror(const char *s)                                       <stdio.h>
int     printf(const char *format [, argument, ... ])               <stdio.h>
int     putc(int c, FILE *stream)                                   <stdio.h>
int     putch(int ch)                                               <conio.h>
int     putchar(int c)                                              <stdio.h>
int     puts(const char *s)                                         <stdio.h>
int     putw(int w, FILE *stream)                                   <stdio.h>
int     _read(int handle, void *buf, unsigned len)                  <io.h>
int     read(int handle, void *buf, unsigned len)                   <io.h>
int     remove(const char *filename)                                <stdio.h>
int     rename(const char *oldname, const char *newname)            <stdio.h>
void    rewind(FILE *stream)                                        <stdio.h>
int     scanf(const char *format [, ... ])                          <stdio.h>
void    setbuf(FILE *stream, char *buf)                             <stdio.h>
int     setftime(int handle, struct ftime *ftimep)                  <io.h>
int     setmode(int handle, int amode)                              <io.h>
int     setvbuf(FILE *stream, char *buf, int type, size_t size)     <stdio.h>
        sopen(path, access, shflag, mode)                           <io.h>
int     sprintf(char *buffer, const char *format [, argument, ... ])  <stdio.h>
int     sscanf(const char *buffer, const char *format [, address, ... ])  <stdio.h>
int     stat(char *path, struct stat *statbuf)                      <sys\stat.h>
char    *_strerror(const char *s)                                   <string.h, stdio.h>
char    *strerror(int errnum)                                       <stdio.h>
long    tell(int handle)                                            <io.h>
FILE    *tmpfile(void)                                              <stdio.h>
char    *tmpnam(char *sptr)                                         <stdio.h>
int     ungetc(int c, FILE *stream)                                 <stdio.h>
int     ungetch(int ch)                                             <conio.h>
int     unlock(int handle, long offset, long length)                <io.h>
int     vfprintf(FILE *fp, const char *format, va_list arglist)     <stdio.h>
int     vfscanf(FILE *stream, const char *format, va_list arglist)  <stdio.h>
int     vprintf(const char *format, va_list arglist)                <stdio.h>
int     vscanf(const char *format, va_list arglist)                 <stdio.h>
int     vsprintf(char *buffer, const char *format, va_list arglist) <stdio.h>
int     vsscanf(const char *buffer, const char *format, va_list arglist)  <io.h>
int     _write(int handle, void *buf, unsigned len)                 <io.h>
```

Rutine de interfaţare (DOS, BIOS, 8086)

```
int     absread(int drive, int nsects, int lsect, void *buffer)     <dos.h>
int     abswrite(int drive, int nsects, int lsect, void *buffer)    <dos.h>
int     bdos(int dosfun, unsigned dosdx, unsigned dosal)            <dos.h>
int     bdosptr(int dosfun, void *argument, unsigned dosal)         <dos.h>
int     bioscom(int cmd, char abyte, int port)                      <bios.h>
int     biosdisk(int cmd, int drive, int head, int track, int sector,
                 int nsects, void *buffer)                          <bios.h>
int     biosequip(void)                                             <bios.h>
int     bioskey(int cmd)                                            <bios.h>
int     biosmemory(void)                                            <bios.h>
int     biosprint(int cmd, int abyte, int port)                     <bios.h>
long    biostime(int cmd, long newtime)                             <bios.h>
void    ctrlbrk(int (*handler)(void))                               <dos.h>
void    disable(void)                                               <dos.h>
int     dosexterr(struct DOSERROR *eblkp)                           <dos.h>
void    enable(void)                                                <dos.h>
unsigned FP_OFF(void far *p)                                        <dos.h>
unsigned FP_SEG(void far *p)                                        <dos.h>
```


Turbo C++ 1.0
Memento - funcţii


```
char    *ecvt(double value, int ndig, int *dec, int *sign)          <stdlib.h>
void    _exit(int status)                                           <process.h, stdlib.h>
void    exit(int status)                                            <process.h, stdlib.h>
char    *fcvt(double value, int ndig, int *dec, int *sign)          <stdlib.h>
void    free(void *block)                                           <alloc.h, stdlib.h>
char    *gcvt(double value, int ndec, char *buf)                    <stdlib.h>
char    *getenv(const char *name)                                   <stdlib.h>
char    *itoa(int value, char *string, int radix)                   <stdlib.h>
long int labs(long int x)                                           <math.h, stdlib.h>
void    *lfind(const void *key, const void *base, size_t *num,
               size_t width, int (*fcmp(const void *, const void *))  <stdlib.h>
void    *lsearch(const void *key, const void *base, size_t *num,
                 size_t width, int (*fcmp(const void *, const void *))  <stdlib.h>
char    *ltoa(long value, char *string, int radix)                  <stdlib.h>
void    *malloc(size_t size)                                        <alloc.h, stdlib.h>
int     putenv(cont char *name)                                     <stdlib.h>
void    qsort(void *base, size_t nelem, size_t width,
              int (*fcmp)(const void *, const void *))              <stdlib.h>
int     rand(void)                                                  <stdlib.h>
void    *realloc(void *block, size_t size)                          <stdlib.h>
void    srand(unsigned seed)                                        <stdlib.h>
double  strtod(const char *s, char **endptr)                        <stdlib.h>
long    strtol(const char *s, char **endptr, int radix)             <stdlib.h>
void    swab(char *from, char *to, int nbytes)                      <stdlib.h>
int     system(const char *command)                                 <process.h, stdlib.h>
```

Rutine de afişare text în ferestre

```
void    clreol(void)                                                <conio.h>
void    clrscr(void)                                                <conio.h>
void    delline(void)                                               <conio.h>
int     gettext(int left, int top, int right, int bottom, void *destin)  <conio.h>
void    gettextinfo(struct text_info *r)                            <conio.h>
void    gotoxy(int x, int y)                                        <conio.h>
void    highvideo(void)                                             <conio.h>
void    insline(void)                                               <conio.h>
void    lowvideo(void)                                              <conio.h>
int     movetext(int left, int top, int right, int bottom,
                 int destleft, int desttop)                         <conio.h>
void    normvideo(void)                                             <conio.h>
int     puttext(int left, int top, int right, int bottom, void *source)  <conio.h>
void    textattr(int newattr)                                       <conio.h>
void    textbackground(int newcolor)                                <conio.h>
void    textcolor(int newcolor)                                     <conio.h>
void    textmode(int newmode)                                       <conio.h>
int     wherex(void)                                                <conio.h>
int     wherey(void)                                                <conio.h>
void    window(int left, int top, int right, int bottom)            <conio.h>
```

Rutine pentru dată şi oră

```
char    *asctime(const struct tm *tblock)                           <time.h>
char    *ctime(const time_t *time)                                  <time.h>
double  difftime(time_t time2, time_t time1)                        <time.h>
long    dostounix(struct date *d, struct time *t)                   <dos.h>
void    ftime(struct timeb *buf)                                    <sys\timeb.h>
void    getdate(struct date *datep)                                 <dos.h>
void    gettime(struct time *timep)                                 <dos.h>
struct tm *gmtime(const time_t *timer)                              <time.h>
struct tm *localtime(const time_t *timer)                           <time.h>
time_t  mktime(struct tm *t)                                        <time.h>
void    setdate(struct date *datep)                                 <dos.h>
void    settime(struct time *timep)                                 <dos.h>
int     stime(time_t *tp)                                           <time.h>
size_t _cdecl strftime(char *s, size_t maxsize, const char *fmt,
                       const struct tm *t)                          <time.h>
time_t  time(time_t *timer)                                         <time.h>
void    tzset(void)                                                 <time.h>
void    unixtodos(long time, struct date *d, struct time *t)        <dos.h>
```

Rutine de lucru pentru liste de argumente variabile

```
type    va_arg(va_list ap, type)                                    <stdarg.h>
void    va_end(va_list ap)                                          <stdarg.h>
void    va_start(va_list ap, lastfix)                               <stdarg.h>
```

DR-DOS 5.0 vs. MS-DOS 5.0

Mai multă memorie, utilitare mai bune - astfel se explică preferinţele utilizatorilor pentru DR-DOS. Cu noua versiune MS-DOS 5.0, Microsoft vrea să recupereze.

De mai mult de un an deja, Digital Research are motive de bucurie: DR-DOS e din ce în ce mai bine primit, concurenţa de la Microsoft pare depăşită: tot mai mulţi vînzători de PC-uri oferă sistemul de operare DR-DOS ca dotare de bază.

Şi este de înţeles de ce MS-DOS 4.01 nu poate concura cu DR-DOS 5.0: utilizarea optimizată a memoriei disponibile, utilitarele foarte bune din cel de-al doilea nu-i lasă practic nici o şansă primului. Pentru a recîştiga o parte din piaţă - sau măcar pentru a nu o pierde de tot, dat fiind că "bătrînul" DOS va determina încă şi în continuare destinele multor PC-uri, Microsoft a "ieşit" cu o versiune nouă, refăcută, MS-DOS 5.0. Ceea ce vă putem relata despre acească versiune se referă la versiunea "beta", deci versiunea încă neoficială, lansată pentru testări. Se poate întîmpla deci ca în versiunea finală să mai apară modificări faţă de ceea ce vă prezentăm. Vă vom face cunoscută apariţia versiunii finale de cum vom şti mai mult. (Gurile rele spun că nici nu are sens să porneşti cu 5.0, fiind mai prudent să aştepţi direct 5.01... Vorba rea are oarecare temei: la lansarea versiunii 4.0, au fost inacceptabil de multe erori şi anomalii în sistemul lansat. Chiar dacă ele au fost destul de rapid eliminate prin lansarea versiunii 4.01, imaginea Microsoft a avut mult de suferit. Se încearcă acum o refacere a bunului nume al Microsoft-ului, printr-o fază de testare neaşteptat de lungă - cea mai completă şi mai costisitoare de pînă acum, dacă ar fi să dăm crezare unor declaraţii făcute în acest sens de către reprezentanţi ai Microsoft).

Avantajul major al DR-DOS era pînă acum modul de gestionare al memoriei. Astfel, în DR-DOS, este posibilă "strămutarea" anumitor părţi din sistemul de operare în zona de high-memory. Astfel că dintr-o dată în strîmta zonă a celor 640 kbyte este mai mult loc - utilizatorul are disponibili pentru aplicaţii pînă la 620 kbyte şi chiar mai mult. Un progres imens, avînd în vedere faptul că aplicaţiile moderne sînt dependente de orice bit de memorie operativă. Şi nu numai sistemul de operare poate fi mutat, ci şi drivere, necesare de exemplu pentru utilizarea unui scanner sau a unei cartele fax. Desigur, premisa pentru acest lucru este o memorie extinsă care să corespundă specificaţiilor LIM.

Posibilităţile de încărcare a memoriei superioare în avantajul celei operative sînt aproape nelimitate: drivere, sistemul de operare şi chiar programe rezidente ca "Sidekick" sînt ţinute astfel în memoria principală, fără ca acest lucru să atragă după sine restricţiile uzuale. Niciodată deci, Word 5.0 de exemplu nu mai spune "Insufficient memory ...": doar sînt disponibili peste 600 kbytes de memorie.

Şi totuşi, nu numai acest lucru constituie avantajul DR-DOS. Cei care l-au creat au lucrat cu multă grijă pentru detalii: astfel, s-au născut utilitare care sînt complet compatibile (în jos) cu modelele MS-DOS, dar oferă în plus numeroase posibilităţi şi dispun suplimentar de cîte o funcţie de help proprie (care poate afişa, la cerere, lista parametrilor şi semnificaţiile lor). Ceea ce arată că pînă şi un sistem de operare orientat pe comenzi poate fi "prietenos".

Din furnitură face parte şi o suprafaţă utilizator grafică, care cuprinde aceleaşi funcţii ca cea din MS-DOS 4.01. Deoarece Digital Research dispune oricum de soft comparabil, suprafaţa grafică DOS nu a mai trebuit special făcută: s-a recurs la deja consacratul "GEM", rebotezîndu-l "ViewMax". Acesta nu are toate funcţiile din "GEM", dar în ce priveşte forma de prezentare şi modul de operare nu sînt mari deosebiri. Altfel este la MS-DOS 4.01, unde Microsoft a respectat cu stricteţe prescripţiile SAA ale IBM, creînd o suprafaţă complet nouă (- dar uitînd cîteva funcţii importante pentru ea).

Acest lucru ar urma să se schimbe acum, în MS-DOS 5.0: ca suprafaţă, se zvonea că va fi ceva asemănator cu Windows, (poate chiar o versiune mai veche de Windows), iar în plus, vor fi atîtea utilitare încît producătorii "clasici" de utilitare ca "PC Tools" sau Peter Norton vor fi nevoiţi să-şi schimbe meseria, la apariţia unor programe ca "undelete", "unformat" sau "diskdoctor". Realitatea însă se prezintă mult mai modest: suprafaţa a rămas aceeaşi, cu cîteva inovaţii desigur, iar utilitare noi sînt puţine: "unformat" şi un program de compresie - decompresie. Şi totuşi: chiar dacă nu se vede din prima privire, verificarea capacităţii de memorie disponibile arată ceva mai mult de 630 kbyte! Pînă acum, erau în cel mai bun caz 580 kbyte. Deci aici Microsoft s-a decis să urmeze exemplul concurenţei: se foloseşte memoria suplimentară pentru a descărca memoria operativă de drivere şi părţi ale sistemului de operare.

În afara acestor performanţe îmbunătăţite relative la memoria suplimentară, mare lucru nu s-a schimbat. Nici măcar n-au fost dotate cu funcţii de help diversele utilitare. În continuare lipseşte o protecţie efectivă prin parolă a fişierelor sau directoarelor. În timp ce în DR-DOS datele pot fi protejate la nivelul sistemului de operare, în MS-DOS 5.0 acest lucru este posibil numai în cadrul "shell" - ului, suprafaţa utilizator. La acest lucru se adaugă faptul că parola este înscrisă într-un fişier în care sînt trecuţi şi ceilalţi parametri ai "shell" - ului. La prima vedere, combinaţia de litere este cifrată, dar o privire mai atentă relevă imediat faptul că este vorba de o simplă translaţie în cadrul codului ASCII. Deci nici un fel de piedică pentru profesionişti.

La acest lucru se adaugă faptul că în afara suprafeţei utilizator, protecţia datelor nu funcţionează de loc, datele putînd fi astfel deteriorate sau văzute chiar şi de către începători. La DR-DOS, protecţia este generalizată la nivelul sistemului.

Această listă poate fi cu uşurinţă continuată: cu comanda "type" de exemplu. MS-DOS 5.0 permite afişarea unui singur fişier la un moment dat, pe cînd DR-DOS permite folosirea de specificatori generici (wildcards) (*,?). Pe lîngă aceasta, o "pauză" după fiecare ecran se obţine cu parametrul "/p", pe cînd MS-DOS cere lansarea unei a doua comenzi: "type nume | more". Cine vrea să schimbe configuraţia MS-DOS, trebuie să treacă prin dificultăţile editării fişierelor "autoexec.bat" şi "config.sys". DR-DOS în schimb oferă pentru aceasta un program special de setup, care permite definirea comodă a modului de folosire a memoriei, defineşte cartela grafică în uz sau configurează drivere pentru mouse-ul folosit.

În ce priveşte "depanările", Digital Research a renunţat la un echivalent al "debug"-ului de la Microsoft, scoţînd în schimb la lumină o relicvă de pe vremea CP/M-ului: "sid", adică "symbolic instruction debugger", care îndeplineşte în esenţă aceleaşi funcţii ca şi "debug".

Din furnitura de livrare a DR-DOS face parte şi un program cache, care permite creşterea aparentă a vitezei de lucru a harddisk-ului. MS-DOS nu are nimic în acest sens.

Şi un alt program a suferit modificările necesare: sub DR-DOS, cu "format"harddisk-ul nu mai poate fi formatat. Funcţia de formatare e preluată de "fdisk", care în MS-DOS răspunde numai de partiţionarea harddisk-ului.

Avantaj DR-DOS şi la comenzile de ştergere: cu "delq" este posibilă ştergerea selectivă a unor fişiere. Pentru fiecare fişier, programul pune o întrebare suplimentară la consolă pentru a se asigura că fişierul trebuie într-adevăr şters. Iar "xdel" permite chiar ştergerea de directoare şi subdirectoare.

Comparaţia celor două suprafeţe utilizator grafice nu relevă nici un fel de asemănări. Ambele răspund însă la două necesităţi importante: pe de o parte permit administrarea harddisk-ului, iar pe de altă parte permit lansarea fără probleme a unor aplicaţii DOS. Programele trebuiesc configurate înainte de lansare. Multe din ele, în special utilitare, au nevoie de preluarea unor parametri înainte de lansare. Ambele suprafeţe permit astfel de aranjamente; se poate face ca parametri să fie predaţi în mod automat, sau se solicită o "fereastră" înainte de orice lansare. Pur vizual, Viewmax oferă mai mult decît Shell: în timp ce Shell atribuie oricărui program nou primit în listă un mereu acelaşi simbol grafic, Viewmax oferă o varietate de "icons". Astfel, programele pot fi repede clasate pe grupe, folosind de exemplu totdeauna o pană de gîscă pentru programele de prelucrare a textelor. MS-DOS Shell a mers pe altă idee: Diversele programe nu sînt lansate direct din director, ca la Viewmax, ci după configurare sînt trecute într-o listă de programe, care poate fi împărţită apoi în grupe şi subgrupe. Utilizatorul îşi poate astfel alcătui "dosare" proprii pentru fiecare categorie de programe. Astfel, toate programele ce ţin de contabilitate pot fi grupate în "dosarul contabilitate", în timp ce utilitarele pot fi grupate în "tools".

Nu prea sînt dificultăţi cu nici un program, la lansarea lui de la nivelul suprafeţei, deoarece ambii concurenţi "se retrag" aproape total din memoria principală la lansarea unei aplicaţii DOS, revenind abia la terminarea programului.

Ambele sisteme de operare dispun de editoare orientate ecran. Editorul MS-DOS, care acum în sfîrşit ar putea înlocui de mult învechitul editor "edlin", orientat pe rînduri, lucrează chiar cu meniuri moderne tip "pull-down" şi ferestre. Funcţiile disponibile sînt însă limitate la căutări, înlocuiri precum şi unele funcţii pe blocuri de text. Pe cînd ecranul poate fi configurat după gust de fiecare utilizator. Ceea ce nu înseamnă sfîrşitul pentru "edlin" - cine îl doreşte, îl găseşte în continuare în furnitură.

Editorul din DR-DOS nu are astfel de "briz-brizuri". Cine lucrează în mod curent cu Wordstar nu va avea probleme, deoarece cele cîteva funcţii pe care le oferă pot fi lansate numai prin combinaţii de taste. Pentru răsfăţaţii obişnuiţi cu ferestre şi SAA, acest lucru cere acomodare şi învăţare.

În afară de editor, MS-DOS a mai fost dotat cu un program nou. Dacă mult timp interpretorul de Basic GW-Basic se livra împreună cu sistemul de operare, acum în furnitura de livrare a sistemului (deci fără alţi bani) se găseşte sistemul de programare "Quick Basic" în versiunea 1.0. Este practic un mediu de dezvoltare Basic, destul de performant. Păcat că lipseşte opţiunea de compilare: e vorba deci în continuare de un interpretor "pur-sînge". În schimb, are un debugger aproape profesional şi un help online la introducere. Subprograme pot fi găsite rapid cu ajutorul unor ferestre suplimentare, deasemenea funcţii pot fi inserate via ferestre. Astfel încît cine are nevoie de un program rapid, vrea să se iniţieze în programare sau pur şi simplu are nevoie de un program pentru el, are tot ce-i trebuie în Quick Basic, chiar şi în această versiune.

Relativ la compatibilitate, chiar şi versiunea "de probă" MS-DOS s-a dovedit surprinzător de stabilă: n-au fost probleme cu nici una din aplicaţiile testate, de la "Word" şi "dBase" pînă la "Norton Utilities", deşi aceste programe apelează deseori funcţiile critice pentru lucrul cu harddisk-ul. E încă vie în amintirea utilizatorilor dezamăgirea provocată de MS-DOS 4.01, care, datorită modificărilor în administrarea harddisk-ului, făcuse inutilizabile versiunile curente ale unor utilitare ca "PC Tools" sau "Norton Utilities". Se pare că de data aceasta Microsoft a mizat pe un grad mult mai mare de compatibilitate.

(continuare în pag. 34)

OOP - Modă sau ... ?

Limbajele de programare procedurale sînt înlocuite de cele orientate obiect. E acesta "leacul miraculos", soluţia tuturor problemelor ? Sau trebuie ţinut seama de anumite premise, obligatorii ? Şi de ce OOP ?

Puţini sînt termenii ce fac atîta vîlvă în ultima vreme. OOP - conceptul viitorului, soluţia pentru toate problemele pe care le implică limbajele procedurale ? Sau doar furtună într-un pahar cu apă ?

În mod cert, niciuna din poziţii nu este necondiţionat corectă sau incorectă. Programele dezvoltate după metodele clasice au un dezavantaj major: întreţinerea. Studii efectuate în această direcţie au relevat faptul că pentru întreţinerea unui program se cheltuieşte 60 - 80 % din efortul total depus. Şi nu este singurul dezavantaj: programele dezvoltate după metode clasice se modifică extrem de greu: deseori, pentru a rezolva sarcini uşor modificate, programele trebuiesc rescrise. Apoi, şi gradul de refolosire al modulelor este destul de redus - lucru evident prin lipsa de productivitate în soft faţă de ritmul de dezvoltare al hardului. O soluţie pentru toate acestea se speră a fi limbajele de programare orientate obiect.

Şi, într-adevăr, filozofia ce stă la baza OOP - programatorul să se orienteze după date, nu după prelucrările lor - convinge repede. Căci paralela între OOP şi logica gîndirii umane este desigur tentantă:

Omul gîndeşte prin noţiuni, care desemnează obiecte - iar în OOP se folosesc "clase" care se concretizează în obiecte. Comunicarea umană e folosită ca model şi în OOP - obiectele îşi trimit unul altuia "mesaje".

Obiectul - noţiunea de bază a filozofiei OOP - reprezintă o entitate care conţine atît un set de date cît şi descrierea metodelor pentru manipularea lor.

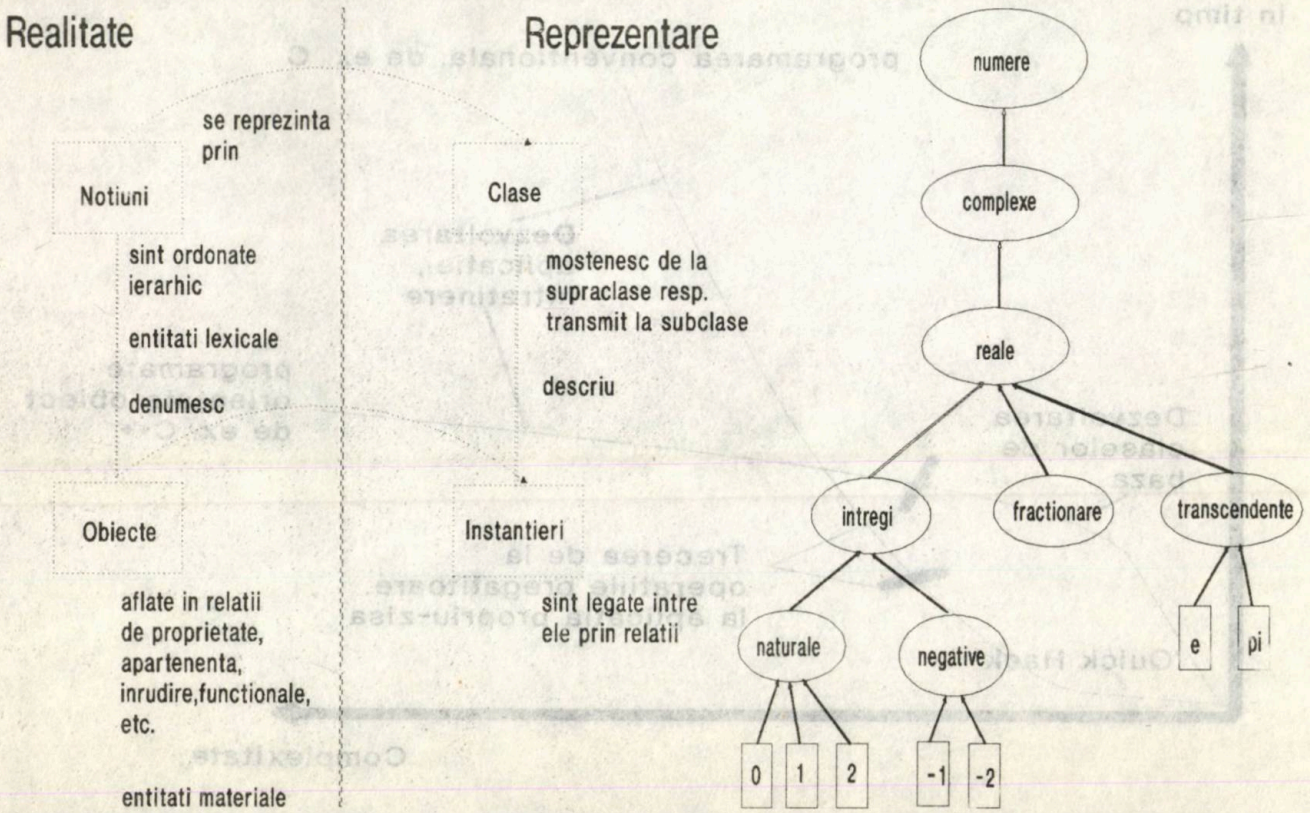
Pentru că obiectele conţin funcţiile necesare manipulării datelor, e suficient să li se adreseze un mesaj corespunzător. Astfel de ex. obiectului "dreptunghi" e suficient să i se trimită mesajul "desenează-te" sau "întinde-te" sau "măreşte-te" - etc. Mesajul nu este deci altceva decît o specificare a unei prelucrări asupra obiectului, care trebuie să precizeze:

-obiectul căruia i se adresează

-operaţia pe care obiectul trebuie să o execute

-parametri actuali necesari operaţiei

Obiectul reacţionează univoc la mesaj; despre polimorfism se vorbeşte cînd obiecte diferite reacţionează în mod diferit la un acelaşi



mesaj. Obiectele sînt descrise în "clase" - este ceea ce gîndirea umană face prin categorisire. Metoda e simplă - proprietăţile comune mai multor obiecte se sintetizează într-o clasă nouă, care conţine toate caracteristicile abstracte ale claselor subordonate. Astfel se constituie supraclasele. Astfel de ex. clasele "autoturism" şi camion ar fi reunite în clasa de bază "maşină".

În afară de aceasta, există instanţierile - obiectele descrise de o anumită clasă. Clasa ar putea să fie "întreg natural" - instanţierile ar fi 1, 2, 3, ...

Avantajele OOP - independenţa reprezentării de implementare, încapsularea, uşurinţa operării modificărilor (de ex. a structurii datelor) etc. - se apropie astfel din ce în ce mai mult de "cipul" soft. Conform acestui concept, munca redundantă (repetată) dispare odată pentru totdeauna, astfel încît rămîn de făcut numai lucrurile realmente noi - ceea ce înseamnă că activitatea programatorului ar consta în a construi aplicaţia do ită din obiecte existente. Ceea ce înseamnă într-adevăr o revoluţie pentru programarea clasică.

Procedeul pare foarte simplu - şi aşa şi este. Totuşi, cîteva dezavantaje majore ale OOP nu pot fi trecute pur şi simplu cu vederea.

Programele OOP nu sînt capabile de multitasking; utilizarea de cod uzual este dificilă; pentru aplicaţii în timp real, există probleme serioase legate de eficienţa codului; programele consumă multă memorie, iar pentru cei implicaţi în proiectare e necesar un efort semnificativ de şcolarizare şi/sau acomodare .

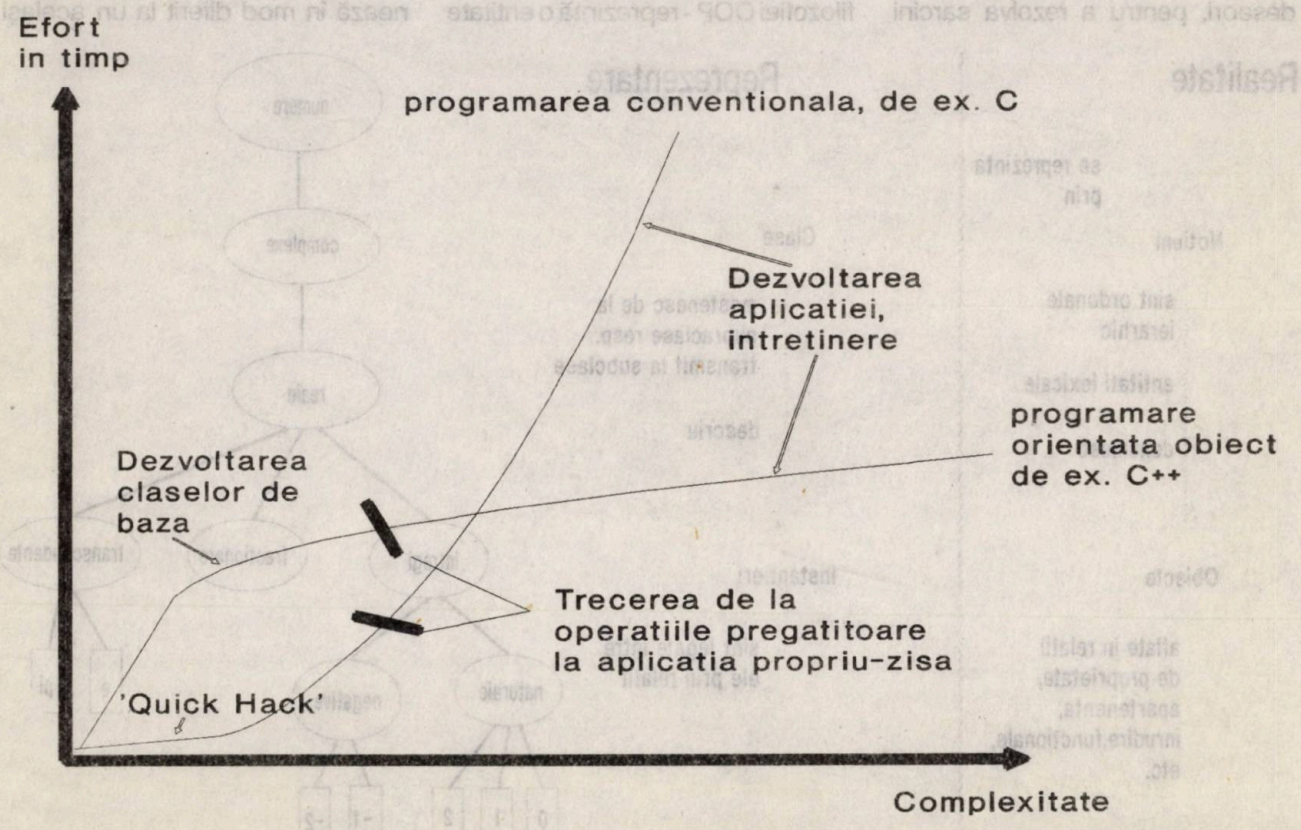
Dezavantajul major este însă în mod evident numai temporar: există încă puţine "cutii de scule" (toolboxes) şi clase gata definite, ceea ce înseamnă multă muncă în faza de proiectare.

Dacă însă clasele de bază ("cipurile soft") sînt definite, se pot într-adevăr selecta componentele potrivite pentru a construi aplicaţia dorită. Citîndu-l pe Georg Heeg: "oricum ar fi, este o mare diferenţă între a învăţa atît la elefant cît şi la pisică sau girafă că gîtul este format din 7 vertebre şi a învăţa acest lucru o singură dată, la nivelul "mamifere".

Pentru că, acum, principala muncă o constituie definirea claselor de bază. Se speră ca în viitor pruductivitatea faţă de programarea clasică să crească între 10 şi 100 de ori. E de ţinut seama de acest lucru; mai ales pentru echipe de proiectare mai mari, investiţia în viitor merită.

Experţii consideră că principalele domenii în care se vor utiliza OOP vor fi suprafeţele utilizator grafice, aplicaţiile tehnice şi sistemele expert, mai puţin bazele de date, procesele numerice şi aplicaţiile în timp real.

(Iosif Fettich)



"C" - Cuceritorul

Ce anume determină atracţia crescîndă exercitată de acest limbaj de programare ? De ce tocmai "C" ?

Cine nu a văzut un calculator ? PC sau mini, pe masă sau în birou - toate au ceva în comun: nu produc nimic dacă nu sînt programate. Ceea ce , desigur, nu vă deranjează, dacă achiziţionaţi o aplicaţie gata scrisă (programată) şi o folosiţi ca atare.

Dar dacă vă prinde microbul ? Vreţi să ştiţi cum anume lucrează calculatorul, aveţi idei noi pentru aplicaţia cu care lucraţi, vreţi să vă scrieţi singur alta - pe scurt, vreţi să programaţi. Primele semne de întrebare: în ce limbă să comunicaţi cu calculatorul ? Primele cercetări vă abandonează într-o adevărată junglă de denumiri: Cobol, Fortran, Algol, Pascal, Assembler, Modula, Lisp, PL/I, C, Ada etc. Ce să alegeţi - şi de ce ?

Vi se recomandă Basic, ca fiind extrem de uşor de învăţat. Încercaţi - şi într-adevăr, primele succese nu se lasă aşteptate. Repede aveţi o serie întreagă de programe care fac ceea ce v-aţi propus.

Pretenţiile vă cresc acum, scopurile pe care vi le-aţi propus devin mai complexe, începeţi să nu aveţi răbdare cînd calculatorul "gîndeşte" prea mult. Entuziasmul iniţial pentru Basic scade pe zi ce trece: constataţi că programele mai complicate devin imposibil de descifrat la cîteva luni după ce leaţi terminat - şi doar era totul aşa de clar cînd aţi scris rutina asta ! - că unele rutine durează insuportabil de mult, că programul care merge impecabil pe calculatorul D-voastră refuză să facă orice altceva decît să blocheze calculatorul prietenului Dvs., căruia vroiaţi să-i faceţi o demonstraţie - ş.a.m.d. Dezamăgit, căutaţi un limbaj de programare mai bun - de data aceasta un compilator, pentru a obţine viteze de execuţie convenabile.

Cu 3 cerinţe mai clar formulate acum (structurare, viteză de execuţie, portabilitate) intraţi din nou în "pădurea" limbajelor de programare. Sub marcajul "structurat" găsiţi o cărare care vă duce pe lîngă Pascal, Ada, Oberon, Modula-2 şi C. Insistînd asupra vitezei de execuţie a codului produs, rămîn doar Modula-2, Assembler şi C. Iar dacă ţineţi ca programele Dvs. să fie şi portabile - i.e., să poată fi rulate fără modificări majore pe orice calculator, atunci căutarea s-a terminat. Tot ce v-a rămas în plasă este "C"-ul.

Gîndit iniţial pentru calculatoare mari, "C" s-a născut în 1972. Între timp, s-a răspîndit atît de mult încît practic nu are concurent şi aproape orice tip de calculator are (cel puţin) un compilator de C disponibil.

Mulţi din producătorii de soft comercial - destinat vînzării - nu mai folosesc alte limbaje de programare, pentru că astfel nu numai că viteza de execuţie a programelor creşte, dar şi din punct de vedere comercial apar avantaje: la transpunerea pe un alt tip de calculator, un program C poate fi transpus unu la unu, adică fără modificări. Trebuiesc eventual adaptate numai caracteristicile speciale - grafica, intrări/ieşiri. Astfel încît nu este de mirare că un produs-program apare pe piaţă simultan pentru mai multe tipuri de calculatoare. În afară de aceasta, compilatoarele moderne de C generează un cod atît de bun şi de eficient încît în foarte rare cazuri este necesară intervenţia la nivelul limbajuluio de asamblare; pentru a "grăbi" vreo rutină mai înceată.

Entuziasmul peren cu care a fost primit "C"-ul a determinat Institutul American pentru Standardizare (ANSI - American National Standard Institute) să încerce o standardizare a limbajului, pentru a spori portabilitatea şi eficienţa lui. Standardul - terminat în 1988 - se referă atît la caracteristici ce ţin de tehnica programării cît şi la caracteristici sintactice. Cel mai important dintre avantajele folosirii lui - e "standard", deci nu există probleme de portabilitate; cel mai important dintre dezavantaje: anumite "trucuri", care produc cod eficient sau utilizează foarte rafinat anumite structuri specifice unui sistem de calcul, nu sînt permise.

Este remarcabilă "libertatea de exprimare" lăsată programatorului în "C". Rămîne la latitudinea lui să-şi scrie programele structurat, clar, sau să scrie programe supereficiente, dar ilizibile. Deoarece există multe programe şi din a doua categorie, criticii spun că "C" nu este un limbaj de programare de nivel înalt. Ceea ce, desigur, poate fi adevărat dacă programatorul este "nedisciplinat". Dar se poate şi altfel - şi programatorul e liber să-şi urmeze propriile opţiuni. Flexibilitatea deosebită a "C"-ului se datorează şi simplităţii nucleului limbajului. Pentru cine are noţiuni de programare, învăţarea "C"-ului nu ridică probleme - instrucţiunile şi cuvintele cheie sînt foarte puţine. Chiar atît de puţine încît foarte repede poate să apară întrebarea: "bine, bine, dar cum să programez ceva într-un limbaj care nu are nici măcar nişte instrucţiuni de intrare/ieşire ?"

Secretul îl constituie filozofia "C"-ului - toate funcţiile sînt externe limbajului şi sînt accesibile prin biblioteci. Cele mai des folosite funcţii (apeluri sistem) sînt adunate într-o bibliotecă standard care se ataşează tuturor compilatoarelor C. Ea conţine rutine pentru intrări/ieşiri generale, pentru gestionarea memoriei şi tratarea şirurilor. Conţinutul bibliotecii standard e prescris cu stricteţe de standardul ANSI, astfel înît orice compilator "C" care corespunde standardului dispune de funcţii identice (cel puţin ca dotare de bază).

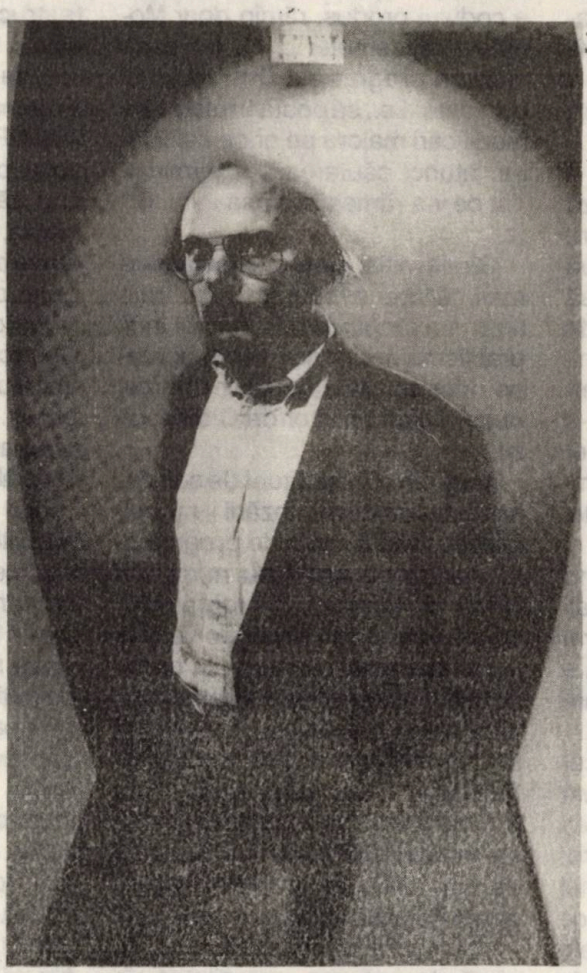
Principiul bibliotecilor de funcţii este foarte avantajos pentru "traducerea" unui program în cod exe-

cutabil (compilarea), deoarece compilatorul nu este încărcat cu numeroase cuvinte-cheie ale limbajului. Operaţiunea de compilare se desfăşoară de aceea semnificativ mai repede la un program C decît la limbajele de programare comparabile. Compilatorul de Turbo Pascal de ex. trebuie să administreze circa 200 de comenzi şi proceduri standard, pe cînd un compilator de C "are de lucru" numai cu 32 cuvinte-cheie şi cîţiva operatori.

Pentru "compunerea" unui program executabil din mai multe module compilate este necesar un alt utilitar: link-editorul. Sigur că şi el are mult de lucru (astfel că o parte din timpul "cîştigat" de C la compilare este pierdut în această fază). Link-editorul, în conformitate cu informaţiile şi tabelele stocate în modulele obiect compilate, caută funcţiile necesare în bibliotecile corespunzătoare şi le adună într-un "tot", un program executabil, de regulă mult mai compact decît la alte compilatoare).

Colecţiile externe de funcţii mai au un avantaj: deoarece o astfel de bibliotecă constă doar dintr-o înşiruire de funcţii C compilate - bineînţeles că şi funcţiile bibliotecii standard sînt formulate în C! - este foarte uşor a extinde acest set de funcţii cu funcţii noi sau a înlocui unele funcţii cu altele mai bune. Astfel că tot "vocabularul" unui sistem C poate fi cu uşurinţă adaptat propriilor nevoi sau dorinţe.

În afară de aceasta, prin "scoaterea" funcţiilordin limbaj, dependenţa de maşină a compilatorului dispare aproape complet, deoarece această dependenţă rămîne limitată la cîteva funcţii de bază. În conformitate cu filozofia Unix, după care orice operaţiune de intrare/ieşire (fie ea la ecran, imprimantă, tastatură sau altceva) este echivalată unei citiri/scrieri într-un fişier, aceste funcţii de bază constau de regulă din cîteva mici rutine pentru a asigura intări/ieşiri la / de la diverse echipamente periferice. Astfel încît la portarea unui compilator C pe un alt calculator nu trebuiesc adaptate decît aceste rutinuţe respectivului sistem de calcul. Prin structura deschisă a "sistemului C", nu numai programele scrise în C sînt uşor portabile, ci şi sistemul în ansamblu poate fi foarte uşor transpus pe alte calculatoare...



Bjarne Stroustrup, "tatăl" C++ - ului

Prin foarte multe din caracteristicile sale, "C"-ul este cel mai "de viitor" limbaj pentru toate tipurile de calculatoare şi, fără nici un dubiu, cea mai bună alegere dacă vreţi să învăţaţi un nou limbaj de programare.

Viitorul, pentru "C", a început deja: criticii au acuzat "C"-ul că lucrează mai ales cu numere şi valori, dar nu sprijină direct prelucrarea obiectelor. Orientarea obiect - care deja este implementată în Pascal şi în Oberon - va duce la o programare completamente abstractă şi independentă de structurile concrete din calculator. Diverse limbaje speciale (mai ales pentru inteligenţă artificială - Prolog, Lisp etc.) urmăresc de mai multă vreme acest obiectiv.

Îmbinarea avantajelor structurale ale "C"-ului cu OOP au născut C++, care se răspîndeşte într-un ritm incredibil. Rămîne de văzut în cît timp C++ va reuşi să detroneze "C" - liderul indubitabil al limbajelor de programare folosite în prezent.

(Iosif Fettich)

DR-DOS 5.0 vs. MS-DOS 5.0

(continuare din pag. 30)

În încheiere, se pune desigur întrebarea care DOS să rămînă pe viitor stăpîn pe valorile depozitate pe harddisk? Privind dinspre confortul utilizatorului, DR-DOS e clar cîştigător. Dar cine nu vrea să renunţe la creaţiile unei din cele mai mari case de soft din lume şi vrea să rămînă într-adevăr 100% compatibil, va lucra şi pe viitor cu MS-DOS. Decizia rămîne de fapt mai degrabă a producătorilor de hardware, dat fiind că ambele sisteme de operare se comercializează numai în domeniul OEM (original equipment manufacturer).

(I.F.)

Dosar: Turbo C++ 1.0

Unul din multele produse foarte reuşite ale firmei Borland, noul mediu de dezvoltare se prezintă cu multe noutăţi. Cele mai importante dintre ele vi le prezentăm în continuare.

În primul rînd, desigur, ++. Corespunzînd în mare măsură "standardului" (neoficial) pentru compilatoarele C++ - este vorba de norma AT&T 2.0 (dar totodată celui mai nou standard ANSI pentru C, cel din decembrie 1988) - Turbo C++ permite programarea cu obiecte "ca la carte". Relativ uşoară este trecerea de la C la C++ nu numai pentru că C++ este un supraset adevărat al C-ului, dar şi pentru că propunerile AT&T au fost preluate între timp de mai toate compilatoarele existente pe piaţă şi disponibilizate în biblioteci adiţionale.

Miezul C++-ului îl constituie noţiunea de "clasă". Cele trei aspecte fundamentale ale programării orientate obiect - încapsularea, moştenirea şi polimorfismul - sînt realizabile, de altfel, şi cu mijloace clasice, dar numai cu preţul unui efort deosebit şi într-un mod destul de "fragil" - e deajuns ca un programator să nu respecte convenţiile stabilite, ca tot "OOP"-ul să nu fie decît o frumoasă faţadă. Nu aşa se întîmplă în C++. Definiţiile de clase permit subdivizări consecvente ale funcţiilor şi datelor, care nu pot fi modificate decît de funcţii interne claselor. Ceea ce duce nu numai la o modularitate mai ridicată, dar uşurează foarte mult depanarea, deoarece erorile de programare care apar pot fi repede urmărite pînă în clasa în care se manifestă.

"Ereditatea" ca principiu fundamental al OOP Înseamnă că din clasele existente se pot genera alte clase, care moştenesc în mod automat toate domeniile de date şi funcţiile clasei vechi, care în mod automat devine parte a clasei noi. Şi în definirea convenţională de structuri se pot concepe construcţii ierarhizate de tipuri de date; în OOP însă, încapsularea se păstrează, mai mult, funcţiile îşi păstrează valabilitatea şi permit în continuare prelucrarea datelor clasei. În C-ul convenţional, ar fi necesare o serie întreagă de "acrobaţii" lingvistice pentru a putea prelucra cu o singură funcţie toate tipurile derivate de date.

Ca o consecinţă directă a "moştenirii" apare polimorfismul funcţiilor. "Polimorfism" vine din greacă şi înseamnă "multe forme": este proprietatea de a da unei funcţii un nume care va fi folosit uniform în toată ierarhia de obiecte, fiecare obiect fiind însă liber să-şi implementeze funcţia în modul cel mai adecvat. Astfel, "moştenirea" duce la preluarea numelui funcţiei, nu neapărat însă şi la cea a implementării. La nevoie, aceasta poate fi refăcută şi duce atunci la înlocuirea codului moştenit. Pe de altă parte, se poate păstra şi vechea versiune a funcţiei. Compilatorul de C++ rezolvă aceste referinţe complexe la momentul execuţiei, parcurgînd ierarhia de obiecte de la obiectul actual şi pînă (cel mult) la rădăcină, pentru a găsi o implementare a funcţiei necesitate. Cel tîrziu la rădăcină funcţia este găsită - altfel, ea nu ar fi putut fi moştenită.

Dar nu toate inovaţiile aduse de C++ sînt legate de programarea orientată obiect. Şi domeniul convenţional este restucturat. Astfel de exemplu biblioteca standard stdio.h este înlocuită cu stream.h. Comentarii de un singur rînd pot fi marcate prin "//". Parametri funcţiilor pot fi transmişi după dorinţă ca valoare sau prin adresă. Etc.

Noutăţi la Turbo-C++ 1.0

- C++ conform specificaţiilor AT&T 2.0
- aritmetică BCD şi numere complexe
- overlay-uri în tehnică VROOM
- mediu de dezvoltare îmbunătăţit
- TEML - macrolimbaj pentru configurarea editorului
- suprafaţă de utilizare operabilă cu mouse
- debugger mult îmbunătăţit
- profiler

Condiţionat de lucrul în C++, Turbo-C introduce în sfîrşit şi calculele cu numere complexe şi aritmetica BCD (cu o precizie de 17 cifre semnificative, necesară în calcule financiar-contabile).

Noutatea care "sare în ochi", pentru cei familiarizaţi cu versiunile mai vechi ale compilatorului, este mediul de dezvoltare, numit IDE (Integrated Development Environment). Noua suprafaţă corespunde standardului SAA şi poate fi folosit (în sfîrşit) şi prin intermediul mouse-ului. Dar cine nu are mouse sau preferă să lucreze cu tastatura, poate să o facă la fel de comod. În afara unor "jucării" atît de plăcute ca ferestrele cu umbre, IDE - ul a suferit numeroase îmbunătăţiri interne. Astfel, pot fi deschise mai multe ferestre simultan, mărimea lor se poate defini, poziţia lor pe ecran se poate fixa după dorinţă. Texte marcate pot fi copiate dintr-o fereastră într-alta prin intermediul clipboard-ului.

Editorul incorporat aduce şi el cîteva noutăţi: se pot edita acum şi surse mai mari de 64 kbyte, sau se pot edita deodată (în paralel) mai multe surse distincte; singura restricţie: dimensiunea totală a surselor editate să nu depăşească 8 Mbyte.

O altă noutate o constituie "TEML" (Turbo Editor Macro Language), care permite reconfigurarea sau modificarea editorului şi a tastelor sau combinaţiilor de taste asociate anumitor comenzi. Adică: dacă în locul editorului standard preferaţi să lucraţi cu un editor ca "Emacs" sau "Vl", prin intermediul macrocomenzilor se poate modifica editorul în aşa fel încît acesta să se prezinte ca cel cu care sînteţi obişnuit. Destul de puternic, acest macrolimbaj permite asocierea unor mici "programe de editare" la anumite taste sau combinaţii de taste, fără a ajunge totuşi la ca-

litățile unui editor specializat (nu există variabile de ex.) Există remediu și pentru acest lucru: editorul cu care lucrați de regulă poate fi înglobat în mediul Turbo, astfel încît să lucreze în ferestrele normale cu care operează acesta, păstrînd în felul acesta modul consecvent în care mediul Turbo-C apare utilizatorului.

Prin intermediul comenzii "asm", se poate insera cod în asamblor, fără a părăsi pentru aceasta mediul C. Pentru aceasta, este necesară însă prezența unui asamblor extern (ca "tasm" sau "masm"). Inserarea de cod în limbaj de asamblare este mult ușurată în felul acesta, deasemenea se simplifică mult predarea parametrilor între funcțiile C și cele scrise în limbaj de asamblare.

Și link-editorul pentru overlay-uri, care disponibilizează tehnica VROOM (Virtuell Runtime Object Oriented Memory Management) în C++, este nou. Altfel decît la tehnicile uzuale, care "descarcă" module nefolosite, lăsînd însă programatorului responsabilitățile legate de interdependența modulelor, VROOM-ul Borland-ului folosește un "dynamic segment swapping" care folosește segmentele ca cele mai mici unități ce se depun pe harddisk, memoria expandată sau în memoria extinsă pentru a face loc în memoria operativă altui fragment de program, care nu a încăput. Cu foarte puține excepții (rutinele de tratare întreruperе de ex.) administrarea dependențelor este asigurată de "managerul de overlay-uri". Programatorul e degrevat.

Deasemenea o noutate este "Profiler"-ul, cuprins pentru prima dată într-o furnitură Borland. Profiler-ul este un soi de cronometror-numărător soft, care poate urmări gestiunea resurselor critice ale calculatorului. Se pot cronometra anumite secvențe de program, se pot număra accesele la disc făcute în anumite secvențe de program sau pe parcursul anumitor rutine, se pot urmări și trage concluzii asupra activității tastaturii, imprimantei, întreruperilor. Astfel, punctele critice ale unui program pot fi mult mai ușor localizate. În același scop poate fi utilă statistica apelurilor, care pentru fiecare rutină sau linie de program contorizează de cîte ori a fost parcursă pe timpul execuției. Informații desigur utile oferă și lista de referințe încrucișate a funcțiilor (care funcție de unde este apelată și ce funcții apelează). Acest "profiling" nu trebuie confundat cu optimizarea internă a compilatorului. Compilatorul poate doar să înlocuiască unele secvențe de cod cu altele, mai eficiente, dar care produc exact același efect. Optimizarea algoritmilor folosiți rămîne în continuare sarcina programatorului, care are acum - prin Profiler - un instrument eficace pentru detectarea părților mai slabe ale unui program.

Multe îmbunătățiri apar și în Turbo-Debugger, prezent acum în versiunea 2.0. În afară de faptul că poate fi utilizat cu mouse, s-a îmbunătățit considerabil modul de administrare al memoriei - debugger-ul nu mai folosește în memoria principală decît 70 kbyte. Tot restul este "depozitat" în memoria extinsă sau cea expandată. Se remarcă în mod deosebit faptul că și debugger-ul este adaptat la OOP, permițînd lucrul cu clase și păstrînd în felul acesta "rotunjimea" acestui mediu C++.

Inovația care în mod cert va cîștiga pe toți cei care s-au luptat cu așa ceva este legată de TSR-uri și drivere. Depanarea acestora este - de regulă - o mare problemă; ce ține de trecut, dacă folosiți noul debugger Borland! Acesta se transformă el însuși într-un TSR, artificiu prin care depanarea de programe rezidente - altfel, "imposibilă", devine relativ confortabilă, ușor de făcut. După ce te-ai obișnuit ca din debugger-ul lansat să instalezi un program rezident (lucru care, după cum bine se știe, nu ar trebui făcut niciodată ! ...) sesiunea de depanare decurge relativ normal:

1. se încarcă Turbo-Debugger-ul
2. se instalează programul TSR de depanat
3. se fixează un breakpoint pe un punct de intrare în TSR

Comparație: Turbo-C++ - MS-C 6.0

Caracteristica	MS-C 6.0	Turbo-C++ 1.0
orientare obiect	nu	da
optimizări relative la UC cu		
8086/8088	da	da
80286	da	da
80386	da	da
80486	nu	nu
optimizări relative la sistemul de operare		
DOS	da	nu
OS/2	da	nu
Windows	da	nu
optimizări relative la codul produs		
ptr. viteză la execuție	da	da
dimensiunea codului	da	da
optimizarea buclelor	da	da
eliminare de cod "mort"	da	da
eliminare expresii parțiale	da	da
răspîndirea constantelor	da	da
biblioteci		
toate modelele de memorie	da	da
80x87	da	da
OS/2	da	nu
Windows	da	nu
debugger		
integrat	nu	da
stand-alone	da	da
profiler		
integrat	nu	nu
stand-alone	nu	da
source-browser		
integrat	nu	nu
stand-alone	da	nu
grafica		
bibliotecă	da	da
drivere	nu	da
CGA	da	da
Hercules	da	da
EGA/VGA	da	da
Super VGA	nu	da*

*Driverele pentru SuperVGA trebuiesc achiziționate separat

4. se face rezident și Turbo-Debugger-ul
5. se activează TSR-ul de depanat la nivel DOS, folosind tasta sau combinația de taste corespunzătoare

Odată activat TSR-ul, acesta se derulează normal, pînă cînd "dă de breakpoint". În acest moment debugger-ul intră pe fir și depanarea se face ca la un program obișnuit.

Această metodă neconvențională, care funcționează

asemănător şi în cazul driver-elor, face din Turbo-Debugger cel mai bun depanator soft disponibil la ora actuală. Nici chiar CodeView-ul Microsoft-ului, în versiunea 3.0, livrat împreună cu Microsoft 6.0, nu ajunge la astfel de performanţe.

Şi încă nu e totul - debugger-ul Borland-ului este în stare să execute pas cu pas un program, **înapoi** !! Te prinde mirarea că astfel de metode nu au fost realizate mai de mult.

Cu toate acestea, nu trebuie uitat că un compilator trebuie, înainte de toate, să producă cod. Şi şi aici, Turbo-C++ nu are motive să se ascundă: chiar dacă C++ produce un overhead destul de substanţial, codul produs de Turbo-C++ încă este semnificativ mai rapid decît cel produs de Quick-C 2.0, fiind destul de apropiat de MS-C 5.1. Calitatea codului produs de MS 6.0 oricum însă nu este atinsă. În ce priveşte optimizările de cod la momentul execuţiei, compilatorul Microsoft este, probabil, imbatabil la ora actuală.

Unul din elementele de strălucire ale oricărui compilator Borland a fost şi rămîne biblioteca grafică. BGI - "Borland Graphic Interface" - lucrează cu drivere ce pot conlucra cu aplicaţii realizate de alte firme, conferind Turbo-C++ - ului o portabilitate fără egal pe sistemele grafice DOS. Cine vrea, poate lucra sub Turbo-C++, fără probleme, cu Super VGA de 1024 x 768 puncte a 16 sau chiar 256 de culori.

Nu trebuie uitat help-ul online oferit de Turbo C++, un ajutor realmente de nepreţuit. Pe o structură tip hipertext, help-ul permite o găsire foarte rapidă a informaţiilor necesitate. Şi nu numai atît: exemplele de utilizare a funcţiilor pot fi copiate din help în programele utilizator, prin intermediul clipboard-ului, apăsînd doar cîteva taste - sau făcind două - trei mişcări şi clic-uri cu mouse-ul.

Mai trebuie amintite şi documentaţiile de utilizare - foarte bune, utilizabile atît de către începători cît şi de către profesionişti.

Ce-i lipseşte Turbo-C-ului pentru a fi totuşi "number one"? Bibliotecile pentru OS/2 şi Windows. Pentru programatorul profesionist care lucrează la aplicaţii ce vor trebui să meargă şi sub OS/2, şi sub Windows, Turbo-C-ul nu este o soluţie. Încă: pentru că deja circulă o versiune de test C++ pentru Windows 3.0. Vremuri grele pentru Microsoft? MS-SDK - pachetul necesar dezvoltării aplicaţiilor sub Windows - devine superfluu; va fi folosit în schimb Whitewater Group Toolkit, care merge numai sub Windows.

La noi, Turbo C++ poate fi achiziţionat de exemplu de la firma Logic, (Bucureşti, 90/75.49.00 sau 75.71.35; Sibiu, 924/4.66.52 sau 4.54.75). Preţurile la zi trebuiesc aflate de acolo; în vest, ele sînt (erau) cam la 450 DM pentru versiunea "normală" (numai compilatorul) şi 690 mărci pentru versiunea "profesională" (compilator + asamblor, debugger, profiler).

Ultima oră: şi Borland C++ 2.0, ("the only complete C and C++ programming environment for DOS and Windows") este disponibil deja; printre alţii, şi la LOGIC, la preţul de 143.055 lei vom mai reveni.

(ing. Iosif Fettich)

Teste de viteză pentru "large model"

	MS-C 6.0	MS-C 5.1	Quick-C 2.0	Turbo-C 2.0	Turbo-C++ 1.0
Array1	30.6	36.0	43.9	51.0	51.0
Array 2	3.0	4.1	38.0	37.4	32.1
Fibonacci	36.5	43.0	53.2	45.7	45.6
Float	36.0	43.0	54.4	58.0	58.0
Integer	0.0	1.9	48.1	48.6	48.6
Loop	0.0	0.0	27.6	27.5	22.1
Sieve 1	20.5	22.5	25.5	25.0	24.2
Sieve 2	16.8	23.0	23.9	23.6	23.5
Whetstone	7.0	6.5	9.0	7.5	7.5

Programe de test de la Microsoft, măsurate pe un AT la 12 MHz

Concurenţii dintr-o privire

Produsul	**Turbo-C++ 1.0**	**MS-C 6.0**	**Topspeed C**
Producător	Borland	Microsoft	Jensen & Co
orientare obiect	da	nu	da
bibliotecă Windows	nu	da+SDK	da+SDK
bibliotecă OS/2	nu	da+SDK	da+SDK
editor	da	da	da
debugger	da	da	da
mouse	da	da	da
linkeditor ptr. overlay-uri	da	da	nu
profiler	în versiunea profesională	nu	în versiunea profesională
preţul (în mărci)	456 684 ptr. versiunea profesională	1585	489 - DOS, 1198, versiunea profesională, 1498 - OS/2

Paradigma Paradox

Înainte să programaţi în PAL (Paradox Application Language), trebuie să înţelegeţi modelul Paradox.

Există vreun utilizator de Paradox pe acolo, pe la Dvs. ? Sau vreun utilizator de Xbase care a auzit o mulţime de lucruri bune despre Paradox şi s-a gîndit să-l exploreze mai îndeaproape ?

Acest articol se adresează ambelor grupuri de utilizatori. Paradox are o creştere considerabilă în acest moment, în special în rîndul corporaţiilor. Neaşteptat Paradox este SGBD-ul (Sistemul de Gestiune a Bazelor de Date) relaţional, pentru PC-uri, cel mai vîndut în rîndul corporaţiilor importante. Un institut de cercetări afirmă că Paradox a acaparat mai mult de 44% din această piaţă, în timp ce dBase III împreună cu dBase IV ating doar 35%, iar Fox şi R:Base au sub 10%. (Datele se referă la S.U.A.)

Borland însăşi pare să aibă un impuls datorat opţiunii client-server SQL (Structured Query Language), unei noi versiuni Paradox Engine, şi posibilităţilor de cuplare cu Quattro Pro 2.0, programul propriu de calcul tabelar.

Dar Paradox a rămas um mister pentru o mulţime de oameni. Mulţi programatori şi utilizatori ai altor SGBD-uri nu înţeleg cu claritate stilul şi modelul Paradox. Paradox are nemeritata reputaţie că ar fi o unealtă de calibru redus recomandabilă doar utilizatorilor finali. De fapt, este un mediu de dezvoltare complet, cu posibilitatea de a genera machete (forme) şi rapoarte, cu un QbE (Query by Example - interogare prin exemplu) pe care toţi doresc să-l copieze şi cu un limbaj puternic care permite proiectanţilor să acceseze, prin intermediul unor construcţii de programare tradiţionale, orice element al mediului interactiv Paradox.

Să înţelegem modelul Paradox

Începătorii în Paradox, adesea, se acomodează greu. Paradox este diferit de alte SGBD-uri tradiţionale; în special utilizatorii de Xbase au probleme de acomodare. Aceasta se întîmplă, în general, din cauză că programele Paradox (script) în loc să facă parte din mediul integrat Paradox, lucrează cu acesta, manipulează tabele în spaţiul de lucru şi navighează prin meniuri. Sub multe aspecte programarea în PAL se aseamănă mai mult cu programarea în limbajul de programare a macrourilor aparţinînd unui program de calcul tabelar, cum ar fi Lotus 1-2-3 sau Quattro Pro, decît cu programarea în dBase sau în clonele sale.

O comparaţie a modelului Paradox cu modelul Xbase ne va ajuta să clarificăm diferenţele.

Modelul Xbase

În nucleul dBase-ului şi al altor produse Xbase stă un mecanism orientat pe articole (Fig. 1). Acest mecanism este proiectat pentru a manipula articole sau linii stocate în fişiere. El le poate indexa, poate căuta anumite valori, executa calcule, face selecţii şi chiar realiza legături rudimentare.

Mecanismul Xbase este optimizat pentru sarcinile pe care le are de îndeplinit. Cînd programatorii în Xbase au nevoie să "înhaţe" un articol, atunci o pot face mai elegant şi mai rapid decît în oricare alt limbaj.

Mecanismul orientat articol este manipulat direct de limbajul de programare Xbase, care este, în toate privinţele, un limbaj tradiţional foarte puternic. El, de asemenea, este optimizat pentru a manipula articole, utilizînd sau nu indexuri, în funcţie de dorinţa utilizatorului sau programatorului.

În exteriorul limbajului de programare Xbase se află interfaţa utilizator. Cei care utilizează interactiv produsele Xbase lucrează direct în această interfaţă, dar de fapt ei nu fac altceva decît să creeze şi să execute mici programe (task-uri) aflate în spatele scenei. În cazul celei mai simple interfeţe, prompterul punct, utilizatorii introduc cîte o singură comandă a limbajului la un moment dat. Din centrul de control al lui dBase IV, atunci cînd utilizatorul selectează generatorul de machete pentru a crea o machetă pentru introducerea de date şi apasă tasta "save", dBase generează sute de linii de cod. Acest cod este interpretat de mecanism pentru a afişa şi manipula date utilizînd această machetă.

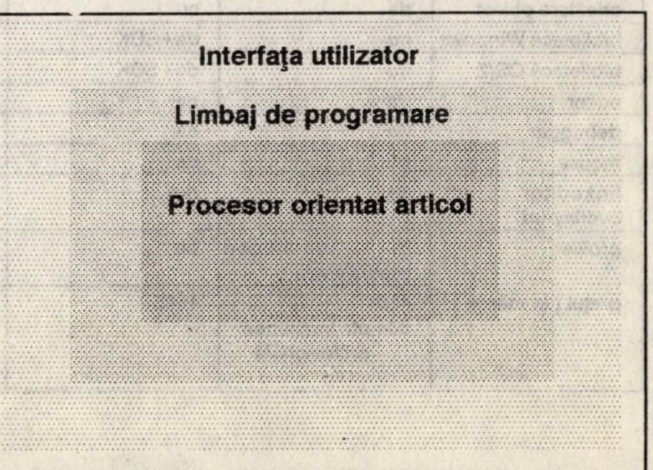


Fig. 1 Modelul Xbase. Limbajul Xbase este situat la mijloc între procesorul de articole şi interfaţa utilizator. Programul manipulează articole la nivelul de jos iar interfaţa utilizator trebuie codificată.

În loc să lucreze cu mediul interactiv, un programator avizat poate crea uşor această machetă în limbajul de programare Xbase, lucrînd cu comenzi familiare.

Din cauză că interfaţa utilizator şi limbajul de programare sînt legate foarte uşor în acest fel şi deoarece interfaţa utilizator se află în exteriorul stratului de cod, este simplu pentru produsele Xbase să aibă interfeţe utilizator diferite. În fapt Ashton-Tate a schimbat interfaţa utilizator a dBase-ului de cel puţin trei ori, iar diferitele clone, cum ar fi FoxPro şi dBXL, îşi au propriile interfeţe utilizator.

Funcţia interfeţei utilizator, în modelul Xbase, este de a crea şi de a manipula cod. Cum se ajunge la nucleul limbajului, fie printr-o interfaţă utilizator (front end), prin intermediul unui limbaj, sau scriind direct într-un editor, practic este lipsit de importanţă.

Modelul Paradox

Modelul Paradox (Fig. 2) diferă de modelul Xbase în trei puncte esenţiale.

În nucleul Paradox-ului, ca şi în cel al dBase-ului, se află tot un mecanism orientat articol. Indiferent de structura sau de modelul SGBD-ului, interfeţei utilizator, sau limbajului de programare, fie că este orientat pe articole sau pe seturi, la un moment dat trebuie manipulate articole individuale stocate în fişiere.

Procesorul orientat pe articole, momentan, este disponibil ca un produs Borland distinct: Paradox Engine. Acest produs este destinat programatorilor în C, C++, Pascal sau Windows care doresc să realizeze aplicaţii cu baze de date utilizînd formatele fişier Paradox, cu un suport complet pentru indexuri, protecţie prin parolă şi protocoale de blocaj multiutilizator. Paradox Engine nu are capabilităţi de interfaţă utilizator, programatorul trebuind să-şi scrie propria interfaţă sau să utilizeze una dintre bibliotecile puse la dispoziţie.

Orientarea tabelă. Prima zonă majoră în care Paradox diferă de Xbase este mijlocul diagramei. Imediat în exteriorul stratului procesorului de articole se află un procesor orientat tabelă, care lipseşte cu desăvîrşire în dBase.

Paradox este proiectat pentru a manipula tabele, în timp ce dBase a fost proiectat pentru a manipula articole. Din acest punct de vedere Paradox este mult mai închis spre un SGBD relaţional, care este de asemenea orientat tabelă. De ex., spre deosebire de Xbase, Paradox nu stochează un număr de articol ca pe un pointer la o linie dintr-o tabelă, ci fiecare linie este identificată prin valoarea primei ei chei primare.

Procesorul orientat tabelă este optimizat pentru a manipula tabele ca seturi relaţionale. "Inteligenţa" înglobată permite Paradox-ului să selecteze o abordare eficientă la nivelul procesării articolelor care stă la baza tuturor operaţiilor pe seturi.

Această optimizare analizează strategia corectă pentru o operaţie pe seturi de fiecare dată cînd este executată. Optimizarea ţine cont de memoria disponibilă şi de celelalte resurse, mărimea tabelelor, prezenţa indexurilor, şi de o mulţime de alţi factori. O aceeaşi cerere poate fi executată utilizînd abordări diferite de fiecare dată.

În consecinţă Paradox execută comenzile relaţionale la nivel de seturi mai rapid decît majoritatea SGBD-urilor. Acest lucru se întîmplă deoarece Borland a avut abilitatea să implementeze procesorul orientat tabele ca o soluţie curată şi elegantă client-server în noul produs Paradox SQL Link; spre deosebire de dBase, Paradox dispune de un procesor SQL puternic integrat în program.

Odată ce se înţelege orientarea pe tabele a Paradox-ului, există o serie de trucuri care pot fi folosite pentru a simplifica programarea.

Orientarea obiect. Tabelele Paradox sînt un tip specific de obiecte, exact cum sînt şi machetele şi rapoartele Paradox. Mult din programarea în Paradox se rezumă la a manipula obiectele construite în program, a naviga printre tabele, a completa imaginile pentru cereri, a selecta machete şi a lista rapoarte.

Aceasta este a doua categorie de diferenţe între Xbase şi Paradox: în Xbase se consumă mai mult timp manipulînd cod pentru a executa operaţii de bază, care în Paradox sînt implementate ca obiecte predefinite.

Deşi Paradox nu are nici una din facilităţile limbajelor orientate

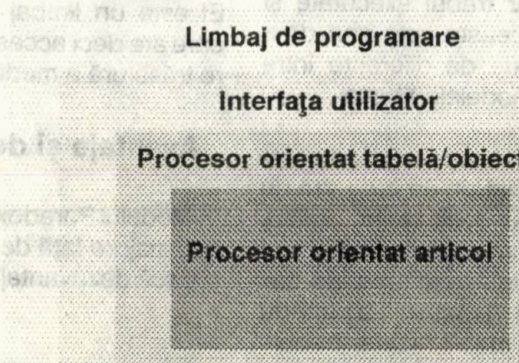


Fig. 2. Modelul Paradox. Modelul Paradox permite PAL să manipuleze construcţii de nivel înalt incluzînd tabele (cu operaţii la nivel de set) şi toate facilităţile interfeţei utilizator

obiect, cum ar fi SmallTalk sau Lisp, el are mai mult dintr-o abordare obiect decît Xbase-ul. Nu există cod PAL nici în machete nici în rapoarte. Acestea există ca şi obiecte binare discrete care pot fi manipulate direct în loc să trebuiască procesate sute de linii de cod.

Interfaţa utilizator. Al treilea nivel în modelul Paradox îl constituie interfaţa utilizator. Aceasta constă din sistemul de meniuri inelar extensiv, incluzînd şi capabilităţile query by example, editorul de machete şi de rapoarte şi diferite module minore, cum ar fi sistemele grafic, de sortare şi de protecţie prin parolă.

Interfaţa utilizator Paradox este locul în care se creează obiectele Paradox, incluzînd tabele, indexuri, machete, rapoarte, setări de imagini şi condiţii de validare a datelor. Acestea se realizează interactiv lucrînd cu ajutorul meniurilor sau editoarelor înglobate.

Unde este PAL ?

Cel mai la exterior nivel este limbajul de programare, care se află "deasupra" interfeţei utilizator. Programele Paradox (script) manipulează interfaţa utilizator ca şi cînd ar apăsa tastele corespunzătoare în locul Dvs. A plasa o tabelă în spaţiul de lucru presupune a executa aceleaşi operaţii cînd programaţi, care ar trebui executate şi interactiv. Aceasta este cea de-a treia categorie de diferenţe între Paradox şi modelele Xbase.

În această abordare Paradox este mai apropiat de un program de calcul tabelar decît de un SGBD. Macrourile programelor de calcul tabelar manipulează foaia de calcul, operînd direct în ea. Într-un program de calcul tabelar trebuie să înţelegeţi exact unde se află macroul în foaie sau în structura de meniuri dacă vreţi să păstraţi controlul asupra lui.

În cazul Paradox-ului se întîmplă acelaşi lucru. Pentru a executa cu succes programele Dvs., trebuie să înţelegeţi exact ce situaţie a creat programul Dvs. în interfaţa utilizator şi în spaţiul de lucru.

De ex., adăugînd datele dintr-o tabelă la cele din altă tabelă se schimbă tabela curentă din spaţiul de lucru. Trebuie să fiţi conştient de aceasta şi programul Dvs. va trebui fie să îndepărteze această tabelă din imagine, fie s-o utilizeze în continuare ca tabelă curentă. În mod similar dacă tastaţi `Do-It! [F2]` într-o machetă multitabelară, cursorul sare înapoi în tabela proprietar (master). Interactiv, dacă doriţi să rămîneţi în tabela de detaliu, va trebui pur şi simplu să tastaşi `[F3]` sau `[F4]` pînă cînd ajungeţi din nou în tabela în chestiune. Programul (script-ul) Dvs. va trebui să facă acelaşi lucru.

Această legătură închisă cu mediul interactiv este una dintre raţiunile pentru care oamenii gîndesc uneori că Paradox-ului îi lipseşte un limbaj de programare robust şi puternic şi că deci este imposibil de a se realiza în el aplicaţii dificile.

De fapt PAL este un limbaj de programare complet, permiţînd variabile dinamice, funcţii şi comenzi definite de utilizator, gestiunea memoriei virtuale şi un set complet de comenzi, funcţii şi construcţii aparţinînd programării structurate. El este un limbaj foarte complet care are deci acces deplin la oricare trăsătură a mediului interactiv.

Avantaje şi dezavantaje

Modelul Paradox are două avantaje majore faţă de modelul Xbase şi două dezavantaje relativ minore.

Un avantaj este acela că este foarte uşor să creezi prototipuri în Paradox. Puteţi naviga interactiv printre paşii necesari, să salvaţi cheile tastate şi codul rezultat este *acelaşi* program pe care l-aţi fi putut scrie cu mîna cu ajutorul unui editor. Paradox permite un număr de moduri de a salva ceea ce aţi făcut într-un program, care ulterior poate fi executat sau combinat în programe mai mari şi mai complete.

Al doilea avantaj major al modului în care este proiectat Paradox este acela că Paradox permite crearea unor utilitare de productivitate ridicată care să opereze direct în mediul Paradox. De ex., nu este dificil să se creeze un program mic care să plaseze în spaţiul de lucru valori de validare pentru cîmpul curent. Programul navighează prin toate aceste meniuri pînă cînd găseşte valorile pe care le caută, le salvează în variabile şi le afişează pe ecran.

Un dezavantaj al modelului Paradox îl reprezintă performanţele mai slabe pe care le obţin programele PAL trecînd prin interfaţa utilizator.
Reprezentarea suplimentară a interpretărilor scade viteza de execuţie a unui program. În mod uzual aceasta nu este o problemă deoarece comenzile PAL realizează mult mai mult într-o linie decît programele Xbase.

De ex., comenzile pentru a afişa o tabelă şi a utiliza o machetă predefinită sînt:

```
view     "CLIENTI"
pickform     "1"
```

Toate indexurile sînt deschise şi aplicate în mod automat. Nu este necesar cod (sursă program) pentru a desena macheta, din moment ce Paradox cunoaşte la cel mai de jos nivel cum să interpreteze şi să folosească obiectul binar.

(continuare în pag. 43)

(R.M.)

Static sau dinamic ?

OOP - Partea a treia

Tema principală a acestei a treia părţi a cursului o constituie metodele virtuale. Cu ajutorul lor, în ultima parte a acestui curs, vor fi tratate structurile dinamice.

Metodele prezentate în primele două părţi ale acestui curs au fost în totalitate statice. Aceasta înseamnă că încă din momentul compilării, se rezervă spaţiul de memorie necesar şi se rezolvă cu ajutorul acestor metode toate referinţele.

Facilitatea de a redefini pentru obiecte derivate, metode noi avînd acelaşi nume ca şi cele vechi, permite adaptarea metodelor în funcţie de necesităţi.

Definiţia obiectului Gr_Pozitie din Unit-ul BPunct din listingul LIST13.PAS este o încercare de definire generală a unui obiect grafic. Obiectul derivat din acesta, Gr_Punct, conţine toate metodele necesare pentru iniţializarea, desenarea, ştergerea şi mutarea unui punct.

Această miniierarhie poate fi extinsă după dorinţă. Toate imaginile imaginabile, cum ar fi cercuri, pătrate sau elipse pot fi derivate din Gr_Pozitie. Pentru fiecare din aceste obiecte grafice există metoda Deseneaza. Dar sub acelaşi nume în funcţie de caracteristicile imaginii grafice se vor ascunde desigur implementări diferite.

Se poate spune şi că fiecare obiect grafic posedă caracteristica de a se putea desena singur pe ecran. Cuvîntul Deseneaza este utilizat pentru a desemna toţi algoritmii posibili de desenare. Acest lucru este uşor de înţeles. Dacă din obiectul Gr_Punct este derivat, de ex., obiectul Gr_Cerc, atunci metodele **Init** şi **Deseneaza** vor trebui reprogramate, fără îndoială. Un cerc necesită ca parametru pentru iniţializare şi raza. Rutina de desenare a unui punct este alta decît cea pentru desenarea unui cerc.

Daca se programează static atunci şi metodele Muta_Abs sau Muta_Rel trebuie redefinite, cu toate că secvenţa de cod nu se modifică. Acest lucru nu s-ar întîmpla dacă în momentul compilării, adresele rutinelor **Sterge** şi **Deseneaza** ar rămîne deschise. Atunci ar exista o posibilitate ca în momentul execuţiei să se stabilească metodele **Sterge** şi **Deseneaza** care să fie folosite. Tocmai această posibilitate o oferă programarea orientată obiect. Turbo-Pascal, începînd din versiunea 5.5, sprijină mecanisme pentru tratarea metodelor virtuale.
{Listing LIST13.PAS}

Pentru o mai bună înţelegere vom mai numi odată diferenţele dintre metodele statice şi cele virtuale. În cazul metodelor statice, în momentul compilării se stabileşte exact ceea ce trebuie să facă. În cazul metodelor virtuale decizia asupra a ceea ce trebuie făcut rămîne deschisă.

Cum funcţionează aceasta ? Unit-ul Gr_Obj conţine noile definiţii. Cuvîntul rezervat **Procedure** înaintea declaraţiilor Punct.Init şi Cerc.Init este înlocuit cu cuvîntul **Constructor**. Un constructor este un antet special de procedură, care rezolvă cîteva iniţializări interne pentru mecanisme virtuale. El este necesar deci pentru a putea lucra cu metode virtuale. Înafară de aceasta constructorul trebuie apelat neapărat înaintea primului apel al unei metode virtuale. El stabileşte referinţele necesare cu ajutorul aşa numitei **VMT** (Virtual Methods Table - tabela metodelor virtuale). În VMT pe lîngă mărimea obiectelor sînt conţinuţi şi pointeri la codul metodelor obiectului.

Dacă nu este apelat Constructor, în locul de ramnificaţie spre o metodă virtuală se află o altă adresă. Urmarea poate fi o blocare a sistemului.

Compilatorul nu are nici o posibilitate de a verifica succesiunea apelurilor metodelor. Este deci sarcina programatorului de a rezolva acest lucru. Singurul ajutor aflat la dispoziţie în Turbo_Pascal este comutatorul {$R+}.

Acesta, gîndit de fapt pentru o verificare a domeniilor, verifică la toate metodele virtuale starea de iniţializare a instanţelor. în cazul în care anumite iniţializări lipsesc, în timpul execuţiei, va fi dat un mesaj de eroare domeniu şi execuţia programului va fi întreruptă. Acest comutator este foarte util în faza de dezvoltare a unei aplicaţii. După testarea programului, comutatorul {$R+} poate fi îndepărtat din nou, programul devenind astfel mai rapid.

Dacă într-o ierarhie de obiecte o metodă este declarată odată virtuală, atunci toate metodele avînd acelaşi nume, ale acestei ierarhii, trebuie să fie de asemenea virtuale. O metodă statică nu poate înlocui niciodată o metodă virtuală. Auxiliarul **Virtual** trebuie scris deci în declaraţiile metodelor pentru toate metodele cu acelaşi nume ale unei ierarhii de obiecte.

Listingul LIST13.PAS clarifică avantajele pe care le aduce utilizarea metodelor virtuale. Obiectul Punct defineşte toate metodele care sînt necesare pentru manipularea unui obiect grafic. Pe lîngă Constructor pentru iniţializare sînt implementate şi rutinele pentru desenarea, ştergerea şi mutarea unui punct. Oricare altă imagine grafică, fie ea un cerc, un unghi drept sau o curbă oarecare, necesită o metodă **Deseneaza** specială pentru reprezentarea ei, dar şi o metoda specială **Sterge**. Metodele pentru mutarea absolută şi relativă însă nu mai trebuie definite şi pentru urmaşi.

Obiectul **Punct** conţine, să zicem aşa, o metodă mamă pentru mutarea tuturor obiectelor grafice care sînt urmaşii obiectului Punct. Pentru acest scop în rutina **Punct.Muta_Abs** şi **Punct.Muta_Rel** numele metodelor **Deseneaza** şi **Sterge** nu conţin nici o referinţă la un obiect anume. Astfel de metode, cum sînt cele două pentru mutarea obiectelor grafice, sînt desemnate în programarea orientată obiect ca metode abstracte. Ele utilizează metode virtuale. În acest caz se va hotărî abia în momentul execuţiei programului care obiect anume trebuie mutat. {Listing GR_OBJ.PAS}.

În partea a doua a acestui curs vom arăta că obiectele pot fi tratate dinamic în acelaşi mod în care pot fi tratate şi variabilele. Unit-ul Gr_Obj demonstrează cum pot arăta astfel de declaraţii. Tipurile de pointeri Gr_PozitiePtr, Gr_PunctPtr şi Gr_CercPtr permit crearea şi îndepărtarea unor obiecte din heap (memoria disponibilă). Pentru acest scop procedurile Turbo-Pascal **New** şi **Dispose** au fost astfel extinse încît să se poată referi şi la obiecte. Pentru utilizarea lui New există două posibilităţi:

```
New (UnCerc);
New (UnCerc,Init(100,100,50));
```

În ambele cazuri se rezervă suficientă memorie în heap pentru a permite o instanţiere a acestui obiect. Apoi **New** transmite înapoi adresa acestei zone de memorie în pointer. În al doilea caz rezervarea de memorie este făcută de constructorul **Init**. Avantajul celei de-a doua metode este la îndemînă. În acest caz odată cu rezervarea spaţiului de memorie necesar, este apelat simultan şi constructorul Init, şi ne asigurăm în acest fel că iniţializările necesare vor fi efectuate. Înafară de aceasta programatorul economiseşte o instrucţiune, deoarece în cazul primei versiuni mai trebuie declarat şi constructorul.

Ca şi toate celelalte tipuri de date tradiţionale, care sînt depuse în heap, şi obiectele pot fi îndepărtate din memorie cu ajutorul procedurii **Dispose**. Dar nu întotdeauna apelul lui **Dispose** este cea mai bună soluţie. Un obiect individual poate conţine pointeri la alte structuri de date sau obiecte, care ar trebui eliberaţi într-o anumită ordine. În acest caz este de bun simţ ca această eliberare a memoriei să fie cuprinsă într-o singură metodă.

Turbo-Pascal oferă această metodă prin intermediul lui **Destructor**. Cuvîntul rezervat Destructor înlocuieşte cuvîntul **Procedure**. Apelul lui **Destructor** uneşte eliberarea zonei de memorie rezervate cu toate celelalte operaţiuni de "curăţenie" necesare.

Cu aceasta devine evident că destructorii acţionează asupra obiectelor dinamice. Ei îşi găsesc întrebuinţarea atunci cînd din heap trebuie îndepărtate obiecte polimorfe. Un pointer la Cerc poate referi un anumit număr de pointeri de tipul Punct. În general dimensiunile obiectelor individuale diferă, deci în momentul compilării nu se cunoaşte mărimea zonei de memorie ce va trebui rezervată. Aceasta explică de ce utilizarea individuală a lui Dispose este interzisă.

Un destructor consultă VMT-ul instanţei apelante. Acolo este memorată dimensiunea în bytes a obiectului. O eliberare reală a memoriei nu se obţine însă decît atunci cînd Destructor este apelat ca o extensie a procedurii Dispose.

```
Dispose (UnCerc,Done)
```

În acest caz destructorul Done este prelucrat ca o metodă normală. După ce instrucţiunea End este atinsă, dimensiunea instanţei este transmisă procedurii Dispose care preia munca de eliberare propriu-zisă. Metoda Destructor însăşi poate să fie goală, eliberarea memoriei se produce totuşi. Ea se realizează practic în fundal.

```
Destructor Done;Virtual;
Begin
End;
```

Exemplul de program "OOP_Incercare Virtuala" scoate în evidenţă mecanismele virtuale. Toate declaraţiile de obiecte sînt plasate în Unit-ul Gr_Obj. Variabilele pointer "UnPunct", "UnCerc" şi "OLinie" pot indica instanţe ale acestui tip de obiect. Ele trebuie deci create cu New. În liniile:

```
New (UnPunct,Init(200,200));
New (UnCerc,Init(100,100,50)); şi
New(OLinie,Init(300,100,150,150));
```

este repetată sintaxa extinsă a procedurii **New**. Cu ajutorul acestor instrucţiuni se rezervă spaţiul necesar de memorie pentru instanţe şi în acelaşi timp este apelat constructorul corespunzător. Apelul virtual propriu-zis este realizat prin sarcina Muta_Rel a instanţei "UnCerc" din linia:

```
UnCerc^.Muta_Rel(50,50);
```

Pentru a putea urmări exact procesul trebuie utilizat modul "Trace Into" al mediului integrat de dezvoltare. Deoarece tipul de obiect Gr_Cerc nu are definită o metodă proprie pentru mutare relativă este utilizată metoda moştenită de la obiectul Gr_Punct. În rutina **Gr_Punct.Muta_Rel** sînt prezente metodele **Sterge** şi **Deseneaza**. În acest caz se stabileşte în momentul execuţiei dacă vor fi utilizate metodele **Sterge** şi **Deseneaza** ale lui Gr_Punct, ale lui Gr_Cerc sau ale lui Gr_Linie. În exemplu sînt utilizate cele ale lui Gr_Cerc.

Primele trei părţi ale acestui curs pot fi rezumate în cele ce urmează: Obiectele sînt tipuri de date definite de utilizator, care înglobează structuri de date şi metodele corespunzătoarte lor. Accesul la date pe o altă cale decît prin intermediul metodelor nu este posibil. Datele sînt deci încapsulate. Tipurile de obiecte pot lăsa moştenire altor obiecte toate caracteristicile lor. În acest mod pot fi construite structuri ierarhice arborescente.

Metodele moştenite pot fi declarate din nou cu acelaşi nume în obiectele fiu. Un obiect fiu poate utiliza atît noile metode definite cît şi pe cele vechi. Utilizarea aceloraşi nume pentru desemnarea unor metode diferite într-o ierarhie este denumită polimorfie.

Obiectele, ca şi variabilele, pot fi tratate şi dinamic. În acest scop procedurile New şi Dispose au fost extinse pentru a putea opera asu-

pra obiectelor. Turbo-Pascal susţine mecanismele pentru metode virtuale. Utilizînd aceste metode se decide abia în momentul execuţiei programului ceea ce este de făcut. Înainte de a utiliza mecanisme virtuale acestea trebuie iniţializate cu ajutorul constructorilor. La eliminarea din memorie a obiectelor dinamice se recomandă utilizarea destructorilor.

Cu aceasta fundamentele programării orientate obiect au fost enunţate. În cursul următor vom clarifica modul de aplicare al acestor tehnici prin intermediul cîtorva exemple.

Metode virtuale

Ele sînt declarate tot la fel ca şi metodele statice, doar că se adaugă cuvîntul Virtual. Toate metodele care vor purta acest nume vor trebui declarate de asemenea virtuale. Ele prezintă faţă de metodele statice avantajul că abia în momentul execuţiei se decide care metodă anume trebuie utilizată. Acest lucru este dependent de tipul instanţei apelante.

Constructor

este un cuvînt rezervat, care înlocuieşte cuvîntul rezervat Procedure. O rutină constructor iniţializează mecanismul metodelor virtuale şi trebuie apelată înainte de apelul lor. El stabileşte o legătură între instanţa care apelează constructorul şi VMT-ul obiectelor.

Virtual Methods Table (VMT)

memorează dimensiunea obiectelor şi conţine pentru fiecare metodă virtuală un pointer la codul ei.

Destructor

Un destructor este o metodă specială, pentru eliminarea de obiecte din memorie şi eliberarea memoriei. Conlucrînd cu procedura extinsă Dispose, zona de memorie ocupată de un obiect poate fi eliberată pînă la ultimul byte. Eliberarea propriu-zisă se petrece în fundal, astfel încît un destructor îşi îndeplineşte funcţia chiar şi atunci cînd nu conţine nici o instrucţiune.

(R. M.)

Paradigma Paradox

(continuare din pag. 40)

În spatele scenei, aceste două comenzi execută mai multe acţiuni:

`[F10]` afişează meniul
`"view"` selectează această opţiune a meniului
`CLIENT` tastează numele tabelei în dreptul prompterului
`[Enter]` acceptă intrarea
** tabela apare pe ecran
`[F10]` afişează meniul
`"Image"` selectează această opţiune a meniului
`"Pickform"` selectează această opţiune a meniului
`"1"` selectează această machetă din lista machetelor disponibile
** imaginea de pe ecran se modifică de la tabelă la machetă.

Performanţele obţinute lucrînd în mediul interactiv sînt îmbunătăţite de un alt punct forte al Paradox-ului, managerul memoriei virtuale, care manipulează eficient aplicaţiile care sînt mai mari decît memoria aflată la dispoziţie. Paradox ajustează dinamic memoria asignată tabelelor, buffer-elor, stivelor şi modulelor programului, în funcţie de resursele hardware disponibile şi de operaţiile executate de către utilizator.

Plasarea în Paradox a limbajului PAL deasupra interfeţei utilizator este restrictivă în alt mod: este dificil pentru Borland să modifice interfaţa utilizator. Din cauză că programele sînt atît de strîns legate de interfaţa utilizator, a o schimba ar însemna să faci aceste programe neutilizabile. Aceasta nu înseamnă că schimbarea este imposibilă ci doar că implică un efort complex şi delicat.

Deci care este concluzia ?

Pentru utilizatorii finali mediile Xbase şi Paradox sînt similare. În fiecare din ele utilizatorul lucrează în contextul interfeţei utilizator fără să ţină cont de faptul că reprezentarea interpretărilor este sau nu prezentă.

Proiectanţii, oricum, *trebuie* să ţină cont de relaţia cu modul de reprezentare. În Paradox, de ex., un programator poate uza de reprezentarea tabelelor pentru a obţine maximum de productivitate. În Xbase trebuie manipulată reprezentarea articolelor pentru a obţine aceeaşi funcţionalitate.

Probabil că cea mai mare diferenţă este aceea că în Paradox programatorul *trebuie* să se plaseze în afara mediului interactiv. Paradox este un program "modal". Atunci cînd navigaţi prin meniuri, selectînd, acţionînd şi manipulînd obiecte, Paradox navighează prin diferite moduri de operare. Fiecare mod este definit de meniul de opţiuni care este disponibil atunci cînd tastaţi `[F10]`.

Anumite acţiuni sînt permise numai în anumite moduri. De ex., unealta de copiere a unei tabele sub un alt nume este apelabilă numai din meniul principal (main). Dacă tastaţi `[Alt-F9]`, (sau determinaţi programul să tasteze `[Alt-F9]`), atunci intraţi în modul `"CoEdit"`, în care acţiunile posibile sînt cele disponibile în acest meniu; copierea unei tabele nu este una dintre aceste opţiuni.

Acest lucru nu este neapărat "mai bun". Avantajele Paradox-ului sînt legate de alte aspecte, cum ar fi creearea de prototipuri şi abilitatea de a manipula date la nivel de set.

Programatorilor Xbase le ia timp pînă să folosească avantajele date de modul în care este proiectat Paradox, de aceea este important să se înţeleagă mediul interactiv şi modelul Paradox înainte de a plonja în PAL.

(R. M.)

```
Unit BPunct;

Interface

Uses Graph;

Type

Gr_Pozitie = Object
  Gr_X, Gr_Y : Integer;
  Procedure Init (X,Y : Integer);
  Function GetPos_X : Integer;
  Function GetPos_Y : Integer;
End;

Gr_Punct = Object (Gr_Pozitie)
  P_Culoare : Integer;
  Procedure Init (X,Y :Integer);
  Procedure Deseneaza;
  Procedure Sterge;
  Procedure Muta_Rel(Xd,Yd:Integer);
  Procedure Muta_Abs
            (X_Nou,Y_Nou:Integer);
End;

Implementation

Procedure Gr_Pozitie.Init
            (X,Y : Integer);
Begin
  Gr_X := X;
  Gr_Y := Y;
End;

Function Gr_pozitie.GetPos_X : Integer;
Begin
  GetPos_X := Gr_X;
End;

Function Gr_Pozitie.GetPos_Y : Integer;
Begin
  GetPos_Y := Gr_Y;
End;

Procedure Gr_Punct.Init
            (X,Y : Integer);
Begin
  Gr_Pozitie.Init (X,Y);
End;

Procedure Gr_Punct.Deseneaza;
Begin
  PutPixel (Gr_X,Gr_Y,GetColor);
End;

Procedure Gr_Punct.Sterge;
Begin
  PutPixel (Gr_X,Gr_Y,GetBkColor);
End;

Procedure Gr_Punct.Muta_Rel
            (Xd,Yd : Integer);
Begin
  Sterge;
  Gr_X := Gr_X + Xd;
  Gr_Y := Gr_Y + Yd;
  Deseneaza;
End;

Procedure Gr_Punct.Muta_Abs
            (X_Nou,Y_Nou : Integer);
Begin
  Sterge;
  Gr_X := X_Nou;
  Gr_Y := Y_Nou;
  Deseneaza;
End;

End.
```

{Listing LIST13.PAS}

```
Program OOP_Incercare_Virtuala;

Uses Crt,Graph,Gr_Obj;

Var
  UnPunct : Gr_PunctPtr;
  UnCerc : Gr_CercPtr;
  OLinie : Gr_LiniePtr;
  Gr_Driver,Gr_Mode : Integer;

Begin
  DetectGraph (Gr_Driver,Gr_Mode);
  InitGraph (Gr_Driver,Gr_Mo-
de,'C:\TP\BGI');
  New (UnPunct,Init(200,200));
  UnPunct^.Deseneaza;
  ReadLn;
  UnPunct^.Muta_Rel(100,0);
  New (UnCerc,Init(100,100,50));
  UnCerc^.Deseneaza;
  ReadLn;
  UnCerc^.Muta_Rel (50,50);
  ReadLn;
  Dispose (UnCerc,Done);
  New (OLinie,Init(300,100,150,150));
  OLinie^.Deseneaza;
  ReadLn;
  OLinie^.Sterge;
  Dispose (Olinie,Done);
  Dispose (UnPunct,Done);
  CloseGraph;
End.
```

{Listing LIST15.PAS}

```
{$R+}
Unit Gr_Obj;

Interface

Uses Crt,Graph;

Type
  Gr_PozitiePtr = ^Gr_Pozitie;
  Gr_PunctPtr = ^Gr_Punct;
  Gr_CercPtr = ^Gr_Cerc;
  Gr_LiniePtr = ^Gr_Linie;

{ aici urmeaza LIST13.PAS incepind
                de la linia 9 }

Gr_Pozitie = Object
  Gr_X,Gr_Y : Integer;
  Constructor Init (X,Y : Integer);
  Function GetPos_X : Integer;
  Function GetPos_Y : Integer;
End;

Gr_Punct = Object (Gr_Pozitie)
  P_Culoare : Integer;
  Constructor Init (X,Y : Integer);
  Destructor Done;Virtual;
  Procedure Deseneaza;Virtual;
  Procedure Sterge;Virtual;
  Procedure Muta_Rel (Xd,Yd : Inte-
ger);Virtual;
  Procedure Muta_Abs
    (X_Nou,Y_Nou : Integer);Virtual;
End;

Gr_Cerc = Object (Gr_Punct)
  Raza : Integer;
  Constructor Init (X,Y,R : Integer);
  Procedure Deseneaza;Virtual;
  Procedure Sterge;Virtual;
End;

Gr_Linie = Object (Gr_Punct)
  XE,YE : Integer;
  Constructor Init
    (X,Y,XEnd,YEnd : Integer);
  Procedure Deseneaza;Virtual;
  Procedure Sterge;Virtual;
End;

Implementation

Constructor Gr_Pozitie.Init
    (X,Y : Integer);
Begin
  Gr_X := X;
  Gr_Y := Y;
End;

Function Gr_Pozitie.GetPos_X : Integer;
Begin
  GetPos_X := Gr_X;
End;

Function Gr_Pozitie.GetPos_Y : Integer;
Begin
  GetPos_Y := Gr_Y;
End;

Constructor Gr_Punct.Init
    (X,Y : Integer);
Begin
  Gr_Pozitie.Init (X,Y);
End;

Procedure Gr_Punct.Deseneaza;
Begin
  PutPixel (Gr_X,Gr_Y,GetColor);
End;

Procedure Gr_Punct.Sterge;
Begin
  PutPixel (Gr_X,Gr_Y,GetBkColor);
End;

Procedure Gr_Punct.Muta_Rel
    (Xd,Yd : Integer);
Begin
  Sterge;
  Gr_X := Gr_X + Xd;
  Gr_Y := Gr_Y + Yd;
  Deseneaza;
End;

Procedure Gr_Punct.Muta_Abs
    (X_Nou,Y_Nou : Integer);
Begin
  Sterge;
  Gr_X := X_Nou;
  Gr_Y := Y_Nou;
  Deseneaza;
End;

Destructor Gr_Punct.Done;
Begin
End;

Constructor Gr_Cerc.Init
    (X,Y,R : Integer);
Begin
  Gr_Pozitie.Init (X,Y);
  Raza := R;
End;

Procedure Gr_Cerc.Deseneaza;
Begin
  Circle (Gr_X,Gr_Y,Raza);
End;

Procedure Gr_Cerc.Sterge;
Var Co : Word;
Begin
  Co := GetColor;
  SetColor (GetBkColor);
  Deseneaza;
  SetColor (Co);
End;

Constructor Gr_Linie.Init
(X,Y,XEnd,YEnd : Integer);
Begin
  Gr_Pozitie.Init (X,Y);
  XE := XEnd;
  YE := YEnd;
End;

Procedure Gr_Linie.Deseneaza;
Begin
  Line (Gr_X,Gr_Y,XE,YE);
End;

Procedure Gr_Linie.Sterge;
Var Co : Word;
Begin
  Co := GetColor;
  SetColor (GetBkColor);
  Deseneaza;
  SetColor(Co);
End;

End.

{Listing GR_OBJ.PAS}
```

Sisteme expert şi asociativitate

Reţele neuronale

Partea a III-a

Sistemele expert acumulează fapte şi reguli. în bazele lor de cunoştinţe; pornind de la acestea şi folosind strategii, ele raţionează.

Cuprinsul cursului:

- Partea 1: Privire de ansamblu asupra temei
- Partea 2: Teoria algoritmului propagării înapoi
- **Partea 3: Sisteme expert şi asociativitate**
- Partea 4: Recunoaşterea scrisului

Sistemele expert simulează cunoaşterea umană în anumite domenii. De aceea se mai numesc şi sisteme bazate pe cunoştinţe. În general sînt compuse din componente diferite. Pe de o parte, baza de cunoştinţe, în care sînt stocate cunoştinţele ce urmează a fi prelucrate (fapte şi reguli). Apoi, maşina inferenţială, folosind strategiile de control preprogramate, "raţionează", emiţînd judecăţi proprii bazate pe datele existente şi cunoştinţele stocate. Pentru o utilizare prietenoasă, acestora li se mai adaugă de regulă o componentă de dialog, care pune întrebări şi preia răspunsurile, o componentă de achiziţie, care "prepară" ştiinţa experţilor pentru a fi preluată în baza de cunoştinţe şi o componentă explicativă, care are menirea de a lăsa să se vadă cum anume ajunge programul să îşi formeze concluziile.

Să considerăm următorul exemplu simplu, care se compune dintr-un fapt şi o regulă. Regula este: "orice om merge în poziţie verticală", faptul este "Bogdan este un om". Se poate concluziona de aici că Bogdan merge în poziţie verticală, deşi acest fapt nu este specificat în mod explicit. El este conţinut implicit în baza de cunoştinţe.

Spre înţelegere, încă două noţiuni. Orice regulă se compune dintr-o parte condiţională şi o parte de acţiune:

DACĂ condiţie ATUNCI acţiune

În exemplul dat, condiţia este "cineva este om" iar acţiunea corespunde "mersului în poziţie verticală". De aceea, regula se mai poate formula şi

DACĂ cineva este om,
ATUNCI el merge în poziţie verticală.

Nu vom intra în amănunte relative la cum se implementează sistemele expert folosind limbaje de programare logică. Ne interesează mai mult care este rolul reţelelor neuronale în acest domeniu.

Este clar că situaţii care pot fi descrise exact cu reguli şi fapte se abordează cel mai bine folosind tehnicile verificate ale programării logice. Dar deja în exemplul dat se poate constata că acest lucru nu este neapărat lipsit de dificultăţi: regula dată nu este general valabilă. Pentru bebeluşi de exemplu, ea nu este adevărată. Şi mai dificilă devine situaţia în care ar trebui prelucrate cunoştinţe nesigure. Nu orice caracteristică poate fi descrisă clar prin două stări (lat sau îngust, lung sau scurt). De cele mai multe ori, ea se manifestă numai în anumite proporţii (merele de exemplu nu sînt 100% roşii sau verzi, ci, de obicei, "şi-şi") sau prin anumite mărimi (1,94m în loc de "înalt").

Cercetarea unor forme adecvate pentru reprezentarea de cunoştinţe nesigure în calculator constituie fundamentul dezvoltării de sisteme IA (inteligenţă artificială).

Acest proces e denumit în general "reprezentarea cunoştinţelor". Se disting în cadrul ei pe de o parte reprezentări simbolice, care pentru prelucrare folosesc tehnici bazate pe reguli logice, şi reprezentări subsimbolice, care apar din modul de lucru al reţelelor neuronale. Care procedeu este mai bun într-un caz particular, trebuie încercat pe aplicaţia concretă.

Cînd şi cum se folosesc însă reţelele neuronale pentru sisteme expert ? Folosirea lor se recomandă atunci cînd partea condiţională a regulilor se compune din mai multe proprietăţi sau cerinţe (de exemplu: "cineva este om" şi "vîrsta lui e mai mare de 2 ani") şi cînd pentru partea de acţiune sînt disponibile mai multe variante (de exemplu "merge aplecat", "merge tîrîş", "aleargă", "sare" etc.). Diversele condiţii ale regulilor se interpretează ca neuroni de intrare ai unei reţele neuronale, unde fiecare proprietate se reprezintă printr-o valoare numerică (de exemplu numărul de ani pentru vîrstă). Variantele posibile ale părţii de acţiune sînt văzute ca şi clase diferite şi li se atribuie neuroni de ieşire diferiţi.

Pentru clarificarea modului de lucru vom prezenta aplicarea metodelor neuronale la un sistem expert exemplu. (*)

Scopul: trebuiesc clasificate nişte obiecte pe baza unor proprietăţi. În afară de aceasta, se doreşte recunoaşterea unor reprezentanţi ai claselor care nu îndeplinesc 100% condiţiile de apartenenţă. Ca obiecte, fie date următoarele tipuri de fructe: cireşe, portocale, mere, pere, prune, banane, struguri, lămîi. Ca proprietăţi

	cireaşă	portocală	măr	pară	prună	banană	struguri	lămîie
are sîmburi	da	da	da	da	da	nu	da	da
coaja e comestibilă	da	nu	da	da	da	nu	da	nu
formă rotundă	da	da	da	nu	nu	nu	da	nu
autohton	da	nu	da	da	da	nu	da şi nu	nu
dulce	da	da	da	da	da	da	da	nu
acru	nu	da	da	nu	nu	nu	da	da
culoare roşie	da	nu	da şi nu	nu	nu	nu	nu	nu
culoare portocaliu	nu	da	nu	nu	nu	nu	nu	nu
culoare albastru	nu	nu	nu	nu	da	nu	da	nu
culoare galben	nu	nu	da şi nu	da şi nu	nu	da	nu	da
culoare verde	nu	nu	da şi nu	da şi nu	nu	nu	da	nu

sînt disponibile: cu sau fără sîmbure, coajă comestibilă, formă rotundă, sînt autohtone, dulce, acru, culoare roşie, culoare portocalie, culoare albastră, culoare verde, culoare galben.

Mai întîi, obiectele cu proprietăţile lor sînt adunate într-un tabel. Se vede imediat că nu este foarte simplu de luat decizii exacte. Merele de exemplu sînt mai degrabă roşii, galbene sau verzi? Deoarece există atît roşii cît şi galbene sau verzi, şi nici una din culori nu este obligatorie, probabil că un "da şi nu" neutru descrie cel mai bine situaţia.

La culori contează numai cele de la fructul copt, astfel că "verde" de exemplu nu intră în discuţie pentru lămîi şi banane. Bazată pe dimensiunea tabelei se construieşte o reţea neuronală care să reprezinte corect proprietăţile şi clasele. Cele 11 proprietăţi sînt interpretate ca 11 celule de intrare. Celor 8 obiecte de clasificat li se asociază cîte o celulă de ieşire, şi anume celula de ieşire nr. 1 îi corespunde obiectului "cireaşă", celula de ieşire nr. 2 obiectului "portocală" ş.a.m.d. Pentru reţeaua neuronală sînt stabilite astfel straturile de intrare şi de ieşire. Reţelele cu propagare înapoi se caracterizează printr-un strat intermediar suplimentar (hidden). Se alege un strat intermediar de 10 celule. Cu ajutorul editorului integrat, se pot produce datele de antrenament pentru reţea. Ele constau din vectori cu 11 componente (cîte una pentru fiecare caracteristică), în cadrul cărora "nu" e reprezentat prin valoarea 0 iar "da" prin valoarea 255. Aici apare o problemă tipică de reprezentare a cunoştinţelor. Cum să fie reprezentat "şi da, şi nu" ca valoare numerică cu care să lucreze reţeaua neuronală ?

Două alternative pot fi luate în considerare: "şi da, şi nu" se poate considera valoare intermediară, ca 128 deci; sau, altfel, se admite atît "da" cît şi "nu". Atunci se stochează două modele diferite, unul cu o componentă nulă pentru "nu" şi unul cu o componentă de valoare 255 pentru "da". În felul acesta se realizează a doua posibilitate, căci "şi da, şi nu" admite ambele cazuri, la urma urmei.

Reţeaua se "antrenează" cu 19 modele de antrenament, care se compun din 8 modele diferite pentru măr, 4 pentru pară, două pentru struguri şi cîte un vector pentru cireaşă, portocală, prună, banană şi lămîie. Obiectivul fazei de antrenament constă în modificarea tuturor ponderilor în aşa fel încît fiecare din cele 19 modele de învăţare să fie clasificat corect. Căci fiecărui vector de intrare îi corespunde un vector de ieşire: astfel de exemplu vectorului de intrare "cireaşă" (255, 255, 255, 255, 255, 0, 255, 0, 0, 0, 0) îi corespunde vectorul de ieşire (1, 0, 0, 0, 0, 0, 0, 0, 0, 0) care reprezintă prima clasă de ieşire.

În acest moment ar putea apare ideea că acest lucru ar putea fi realizat banal, prin simpla stocare a perechilor de antrenament prin intermediul unei tabele indexate. În acest caz, valorile de ieşire nu ar fi calculate, ci s-ar regăsi ca într-un dicţionar. Foarte mici abateri ale "cuvîntului" căutat faţă de cuvîntul memorat ar duce însă la eşecul căutării.

Deoarece valorile de intrare sînt ponderate la reţelele neuronale, numerele individuale contribuie mai mult sau mai puţin la calculul rezultatului, al vectorului de ieşire. De aceea, valori asemănătoare la intrare produc valori asemănătoare la ieşire. Se vorbeşte astfel de capacitatea asociativă a reţelelor neuronale. Modele de intrare incomplete se pot astfel clasifica corect sau completa.

Proprietăţile asociative trebuiesc dezvoltate în faza de învăţare. Este limpede că acest lucru devine cu atît mai dificil cu cît trebuie reprezentaţi mau mulţi parametri de intrare (neuroni de intrare) şi mostre de antrenament. Dacă în plus modele foarte diferite aparţin unei aceleiaşi clase, procesul de convergenţă este îngreunat suplimentar. De aceea, mai ales începătorilor li se întîmplă deseori

ca sistemul să fie divergent, deci învăţarea să nu aibă succes. Ca rezultat al unui proces de convergenţă nereuşit, reţeaua reprezintă modelul pe o celulă de ieşire greşită, neîndeplinindu-şi astfel scopul.

Ca proiectant începător de reţele neuronale, ce-i de făcut? Este iarăşi evident că selecţia regulii de învăţare corecte, mărimea parametrilor potriviţi, structura reţelei (numărul de neuroni) şi reprezentarea cunoştinţelor sînt totdeauna dependente de problemă. Dacă ar exista penru aceasta o regulă general valabilă, atunci aceasta ar fi fost deja de mult implementată în soft. Astfel, toate problemele reprezentabile prin această tehnică ar fi deja rezolvate. Deci este necesară o anumită experienţă în utilizarea acestor tehnici pentru a atinge obiectivele dorite. Ceea ce nu e o noutate, desigur: şi pentru a bate un cui în perete, trebuie să ştii să umbli cu ciocanul, deoarece altfel apar probleme cu degetele.

Pentru a rezolva problema de mai sus a clasificării fructelor, a dus la reuşită regula de învăţare 1, predefinită, cu parametri predefiniţi beta1 = 1.0 şi beta3 = 1.0, toleranţa predefinită T = 850 şi condiţia de abandon 3, care abandonează în mod automat procesul de învăţare cînd pentru fiecare din modelele de învăţare sînt satisfăcute condiţiile de toleranţă impuse.

Deoarece regula de învăţare nr. 1 lucrează pendinte de un generator aleator, mai multe teste diferite au dus la rezultate diferite pentru durata procesului de învăţare.

Pentru a "învăţa" cele 19 modele, sistemul a avut nevoie de 130 pînă la 270 de iteraţii. Ceea ce pe un AT (lent) de 10 MHz corespunde unei durate de 1-2 minute. Nu trebuie uitat însă că un începător are nevoie de un timp semnificativ mai mare pentru a găsi valorile optimale menţionate mai sus.

După terminarea cu succes a fazei de învăţare se poate şi trebuie verificată "capacitatea" reţelei. Ceea ce se face modificînd, cu editorul incorporat, valorile unui model existent sau producînd pur şi simplu modele noi care sînt predate apoi spre evaluare (identificare) reţelei, prin simplă apăsare a unei combinaţii de taste. În felul acesta se pot face rapid şi comod studii particularizate. Pentru verificarea profesională a capacităţilor învăţate se pot testa şi fişiere întregi, în mod automat, şi ieşi cu rezultatele fie la ecran, fie la imprimantă. Se vede că reţeaua "învăţată" e "testată" cu modele necunoscute (ei). Rezultatele nedorite se corectează, preluînd mostrele greşit identificate în fişierul de antrenament şi refăcînd faza de învăţare, pentru a îmbunătăţi astfel performanţele reţelei. Prin aplicarea repetată a combinaţiei învăţare/verificare, prin intermediul setului (fişierului) de date de antrenament, se obţine o apropiere din ce în ce mai mare de caracteristicile optime dorite ale reţelei.

Proprietatea de asociativitate mai sus pomenită a fost testată pe un caz "dur". Se cere antrenarea unei reţele care să recunoască cuvinte tastate greşit. Pentru acest lucru au fost selectate 12 comenzi DOS: CHDIR, CHKDSK, COPY, DEL, DIR, FORMAT, MKDIR, MODE, PATH, PRINT, RMDIR, TYPE. Ca intrare în reţea se foloseşte valoarea corespunzătoare ASCII a literei, deci de exemplu pentru CHDIR vectorul (67, 72, 68, 73, 82). Pentru a îngreuna sarcina reţelei, se permit cuvinte avînd pînă la 7 caractere, astfel că CHDIR se poate scrie >_CHDIR_<, >__CHDIR< sau >CHDIR__< cu vectorii de intrare total distincţi (20, 67, 72, 68, 73, 82, 20), (20, 20, 67, 72, 68, 73, 82), (67, 72, 68, 73, 82, 20, 20). Fiecăruia din cele 12 cuvinte i se asociază o celulă de ieşire a reţelei. Scopul este de a antrena reţeaua în aşa fel încît celula de ieşire corespunzătoare lui CHDIR să se activeze de exemplu şi cînd la intrare este >CHDIT<. În felul acesta pot fi eliminate erorile de tastare. Pentru a putea aprecia modul de punere al problemei, ar trebui ţinut seama şi de asemănarea între trei din comenzile selectate: CHDIR, RMDIR şi MKDIR. Ca fişier de antrenament s-a pornit pentru început cu cîte două variante deplasate pentru >FORMAT< şi >CHKDSK<, respectiv cîte 3 variante pentru celelalte cuvinte, astfel că în total s-au obţinut 34 de modele pentru antrenament. Fişierul de test în schimb conţine toate variaţiile posibile, dar cu o singură literă greşită - în total, 3.856 de cuvinte.

După o învăţare reuşită, cu 16 neuroni ascunşi şi regula de învăţare nr. 1, din cele 3856 de mostre necunoscute, 2170 au fost clasificate corect. Deci din 34 mostre de antrenament se asociază 2170. Prin adăugarea altor mostre în fişierul de antrenament, rezultatul a putut fi îmbunătăţit: cu 123 mostre de antrenament (şi 18 neuroni ascunşi) au fost clasificate corect 2818 cuvinte; cu 198 mostre de antrenament (şi 20 de neuroni ascunşi), 3191.

Pentru memorarea ponderilor necesare la 20 de neuroni ascunşi şi a informaţiilor relative la topologia reţelei sînt necesari numai 4 kbyte de memorie.

Cine îşi face gînduri relativ la numărul mare de mostre de antrenament, va realiza că interesantele proprietăţi ale reţelelor neuronale ar putea fi îmbunătăţite prin combinare cu alte tehnici. Cum anume se face acest lucru, veţi afla în episodul următor.

(I.F.)

(*) Acest curs ar trebui desigur secondat de simulatorul la care se fac referiri. Sperăm că experienţele descrise pot arunca totuşi o geană de lumină asupra domeniului. chiar şi aşa, "pe uscat".(N.red.)

Parolă variabilă

N-aţi fost niciodată în situaţia de a porni sistemul cu un musafir nepoftit în spatele Dvs ? Nu vi s-a întîmplat, mai ales dacî nu tastaţi foarte repede, ca acesta să "prindă" parola de acces şi să trebuiască s-o modificaţi ?

Ei bine, dacă da, vă puteţi rezolva problema în 20 de minute, asamblînd programul ce urmează.

Vă propun o parolă care să fie data curentă în forma ""llzz" pentru varianta prezentată în listing sau "zzll" dacă inversaţi liniile marcate prin comentarii ca linia 1 şi linia 2.

Programul PASSW este foarte strict: trebuie să tastaţi exact 4 cifre corecte. Mai multe caractere tastate (se admit maximum 101) sau un singur caracter greşit duc la blocarea sistemului într-o buclă infinită din care doar reset-ul îl poate scoate - şi atunci o luăm de la capăt.

PASSW nu afişează caracterele tastate, ci doar cîte un punct pentru a facilita controlul introducerii lor. Aţi greşit un caracter ? Nici o problemă: puteţi utiliza liniştit backspace pentru corectarea erorii.

Deci: creaţi fişierul PASSW.ASM cu ajutorul unui editor, apoi, în ordine, executaţi comenzile DOS

> tasm passw
> tlink passw

(sau echivalent, dacă nu lucraţi cu Turbo - Assembler).

Pentru ca PASSW.EXE pe care l-aţi obţinut să se execute automat după pornirea sistemului, trebuie ca fişierul AUTOEXEC.BAT să conţină liniile exemplificate în PASSW.BAT

Dacă totul a mers corect, nu rămîne decît să resetaţi PC-ul şi să introduceţi în ordine data, ora (le puteţi lăsa nemodificate) şi parola. Nu uitaţi: data introdusă va fi cea cerută de parolă - chiar dacă este una falsă. Atunci cînd vă secondează un musafir nepoftit, nu aveţi decît să introduceţi o dată falsă, pentru a fi sigur că nici în acea zi el nu va putea avea acces la PC-ul Dvs. Chiar dacă reuşeşte să urmărească ceea ce tastaţi, parola pe care o va desoperi e valabilă doar o singură zi pe an.

(Daniel Buleu, Iaşi)

```
1       data    segment
2   ED  db  08h, ' ', 08h, '$'
3   EPAR db  ' Parola :$'
4   LOC db  0,0,0,0
5   LC  db  101 dup (0)
6   EAD db  0dh, 0ah, 07h, 07h, 20 dup (' ')
        db  , 'Accesul interzis !!'
7       db  07h, 0ah,'Opreste calculatorul !!$'
8   EOK db  0dh, 0ah, 0ah, 0ah, 'Bine ati venit !!!!$'
9   data    ends
10
11  code    segment
12  assume  cs:code, ds:data,ss:code
13
14  THA     proc
15      mov     ah,0
16      mov     bh,10
17      div     bh
18      add     ax, 3030h
19      mov     LOC [si], al
20      inc     si
21      mov     LOC [si], ah
22      ret
23  THA endp
24
25  start:
26      mov     ax, data
27      mov     ds, ax
28      mov     ah, 9
29      mov     dx, offset EPAR
30      int     21h
31      mov     ah, 2ah
32      int     21h
33      mov     si,0
34      mov     al, dh
35      call    THA
36      inc     si
37      mov     al, dl
38      call    THA
39      mov     si, 0
40  beg:    mov     ah,07
41      int     21h
42      cmp     al,0dh
43      je      e1
44      cmp     al,8
45      jne     e0
46      mov     ah,9
47      mov     dx, offset ED
48      int     21h
49      dec     si
50      jmp     beg
51  e0:     mov     LC [si], al
52      inc     si
53      cmp     si,100
54      je      ad
55      mov     ah,2
56      mov     dl,'.'
57      int     21h
58      jmp     beg
59  e1:     dec     si
60      cmp     si, 3
61      jne     ad
62  e2:     mov     dh, LOC [si]
63      cmp     dh, LC [si]
64      jne     ad
65      dec     si
66      cmp     si, 0
67      jne     e2
68      mov     ah, 9
69      mov     dx, offset EOK
70      int     21h
71      jmp     quit
72  ad:     mov     ah,9
73      mov     dx, offset EAD
74      int     21h
75      jmp     ad
76  quit:   mov     ah,4ch
77      int     21h
78  code    ends
79      end start
80
```

PASSW.ASM

```
date
time
echo off
break off
passw
break on
echo on
```

(în) AUTOEXEC.BAT

Protecţia programelor

Cu filtrul prezentat în cele ce urmează se poate proteja orice program EXE sau COM. Se inserează ca primă instrucţiune în fişier instrucţiunea "ret", ce are codul pe un singur octet (0c3h). Efectul este că la apelul programului se execută acest "ret", astfel că programul se termină şi se întoarce în DOS. Prin această inserare, sistemul de operare este determinat să trateze şi fişierele EXE ca pe cele COM. Prin urmare, fişierele EXE mai mari de 64kbyte care au fost protejate prin MODIF vor anunţa printr-un mesaj de eroare că nu încap în memorie, ceea ce altfel oricum însă nu deranjează. Repetarea comenzii MODIF pentru un fişier îndepărteză protecţia pusă. Fişierul de comenzi prezentat anexat permite protejarea comodă a programelor.

(Computer Persönlich, 7/91)

```
1   uses     Dos;
2   const    Max = $fd00;
3   var      R : registers;
4            b : array [1..Max] of byte;
5   begin
6        R.DX := Ofs(b); R.DS := Seg (b);
7        R.CX := 1;      R.BX := 0;      R.AH := $3f;
8        MsDos (R);
9        if b[1] $c3 then begin
10           b[2] := b[1];
11           b[1] := $c3;
12           R.CX := 2; R.BX := 1;      R.AH := $40;
13           MsDos (R);
14       end;
15       repeat
16           R.CX := Max;
17           R.BX := 0;
18           R.AH := $3F;
19           MsDos (R);
20           R.CX := R.AX;
21           R.BX := 1;
22           R.AH := $40;
23           MsDos (R);
24       until R.AX = 0;
25  end.
```

MODIFY.PAS

```
echo off
modify <%1> modify.tmp
copy modify.tmp %1 > NUL
del modify.tmp > NUL
echo on
```

MODIF.BAT

Reset la imprimantă

PRNRESET poate fi folosit într-o aplicaţie BASIC pentru a produce resetarea imprimantei. Pentru aceasta, programul conţine o scurtă rutină în asamblor, care este introdusă în memorie folosind instrucţiuni POKE. La lansare trebuie specificată imprimanta (portul paralel) ce trebuie resetat.

(Computer Persönlich, 10/91)

Ieşire în DOS

De multe ori în unele aplicaţii am vrea să putem executa comenzi DOS, fără a ieşi din aplicaţia noastră. O alternativă o oferă programul TPDOS.PAS. Ideea programului a pornit de la programul demonstrativ al firmei Borland, EXECDEMO.PAS, livrat odată cu mediul de programare Turbo Pascal 5.5.

Îmbunătăţirea faţă de EXECDEMO.PAS este afişarea prompterului (ca pentru pg). În acest sens am folosit procedura GETDIR(.,.), unde primul parametru este un întreg ce reprezintă unitatea (0 = curentă, 1 = A, 2 = B, ...), iar al doilea parametru este un string ce reprezintă calea. Propun îmbunătăţirea programului DOS.PAS prin interpretarea unei comenzi PROMPT cu sintaxa: PROMPT [$litera], unde prin [ ] am marcat o secvenţă ce se poate repeta.

(Ioan Tiberiu Socaciu, Cluj)

```
1   {$M 8192, 0, 0}
2
3   uses DOS;
4   type comanda = string[127];
5   var ok:      boolean;
6       path:    string[100];
7       drive:   integer;
8       c:       comanda;
9
10  procedure command (c: comanda);
11  begin
12      SwapVectors;
13      Exec (GetEnv ('COMSPEC'), '/C ' + C);
14      SwapVectors;
15      if DosError < > 0 then
16          WriteLn ('Could not execute COMMAND.COM');
17      WriteLn;
18  end;
19
20  begin
21      writeln;
22      writeln ('DOS Simulator');
23      writeln ('(C) 1991 by SOKYSOFT');
24      writeln ('type soky to exit');
25      writeln;
26      ok: = true;
27      drive: = 0;
28      repeat
29          getdir (drive, path);
30          write (path, '>');
31          readln (c);
32          if c < > 'soky' then command (c)
33                          else ok: = false;
34      until not ok
35  end.
```

TPDOS.PAS

```
1   1 DRESET = &HFF00
2   2 DATA &h55, &h89, &he5, &h8b, &h76, &h6, &h8b, &h14
3   3 DATA &hb4, &h1, &hcd, &h17, &h5d, &hca, &h2, &h0
4   4 FOR I=0 TO 15: READ B%
5   5 POKE (DRESET+I), B% : NEXT I
6   10 PRINT  "Reset la imprimanta !"
7   20 INPUT  "Imprimanta (0,1,2): ", A%
8   30 CALL DRESET (A%)
```

PRNRESET.BAS

Există o dischetă în unitate ?

Ideea din următoarea funcţie este că, folosind perifericul NUL al DOS se verifică dacă unitatea de disc este pregătită (cu dischetă) sau nu. Programul este scris în Pascal.

(Computer Persönlich, 10/91)

```
1  program    TestDisk;
2  uses       crt;
3  var        ch: char;
4  const      ESC = #27;
5
6  FUNCTION   E_disc_in_unit ( drive : char) : boolean;
7
8  var
9      f: file; dstr: string;
10 begin
11     E_disc_in_unit : = false;
12     dstr : = drive + ':\NUL';
13     Assign (f, dstr);
14     FileMode : = 0;
15     {I-}
16     Reset (f);
17     {I+}
18     if IOResult = 0
19         then begin
20             E_disc_in_unit : = true;
21             Close (f);
22         end;
23     FileMode : = 2;
24 end;
25
26 begin
27     WriteLn ('Se iese din test apasind <ESC>');
28     repeat
29         ch : = ReadKey;
30         if  E_disc_in_unit ('A')
31         then    WriteLn ('Este discheta in unitatea A:')
32         else    WriteLn ('Nu este discheta in unitatea A:');
33     until ch = ESC;
34 end.
```

TESTDISC.PAS

```
1  #include <conio.h>
2  #include <dos.h>
3  #include <stdio.h>
4
5  void main (void)
6  {
7      char    far * cFP;
8      char    far * cHP;
9      short i;
10
11     FP_SEG (cFP) = FP_SEG (cHP) = 0x4000;
12     FP_OFF (cFP) = FP_OFF (cHP) = 0xff67;
13
14     for (i=0; i<20; i++)
15     {
16         printf ("Far-Pointer: %04x:%04x, "
17             "Huge-Pointer: %04x:%04x\n",
18             FP_SEG (cFP), FP_OFF (cFP),
19             FP_SEG (cHP), FP_OFF (cHP) );
20         cFP + = 0x10;
21         cHP + = 0x10;
22     }
23 }
```

FARHUGE.C

Autoverificare contra viruşilor

Cu ajutorul unităţii SELFCHK ce v-o prezentăm mai jos, un program oarecare Turbo Pascal poate să-şi dea singur seama dacă este sau nu atacat de un virus. Unit-ul, care nu exportă nimic, constă dintr-o parte de iniţializare care depune o sumă de control în program şi o parte de verificare, care la fiecare lansare a programului - excepţînd-o pe prima - verifică faptul că suma calculată coincide cu cea memorată. În caz de diferenţe, rămîne la latitudinea utilizatorului să oprească lucrul sau să ignore pericolul. În caz de coincidenţă, programul se desfăşoară normal. Pentru instalarea unit-ului, este suficientă numirea lui în directiva Uses.

(Computer Persönlich, 7/97)

```
1  unit SelfChk;
2  Interface
3  Uses Crt;
4  Implementation
5  var f          : File;
6      i, citit   : Word;
7      pw, dd     : Array [1..16] of Char;
8      sum        : Array [1..4] of Char;
9      xsum, chksum : LongInt;
10 begin
11     pw : = 'if 5/91 PRACTICA';
12     sum : = #0#0#0#0;
13     xsum : = 0;
14     Assign (f, ParamStr(0)); Reset (f,1);
15     repeat
16         BlockRead (f, dd, 16, citit);
17         for i : = 1 to 16 do xsum : = xsum + Ord(dd[i]);
18     until (citit < 16) or (dd = pw);
19     Blockread (f, chksum, 4);
20     if chksum = 0 then begin
21         WriteLn ('Checksum marcat !');
22         Seek (f, FilePos(f)-4);
23         BlockWrite (f, xsum, 4);
24         Close (f);
25     end else begin
26         Close (f);
27         if (chksum = xsum) and (citit = 16) then begin
28             WriteLn ('Autotest terminat cu succes!');
29         end else begin
30             WriteLn ('Pericol de infectie cu virus.');
31             WriteLn ('Continuare sau Stop (C,S): ');
32             if UpCase (ReadKey) < > 'C' then Halt (1);
33         end;
34     end;
35 end.
```

SELFCHK.PAS

Pointeri FAR şi HUGE în C

Modul de lucru cu segmente de memorie a procesoarelor Intel a dus la împărţirea pointerilor din C în trei categorii: "near", "far" şi "huge". Dacă un pointer "near" este doar un offset de 16 biţi la segmentul curent, celelalte două tipuri sînt pointeri de 32 biţi, constînd din segment şi offset.

Atît pointerii "far" cît şi cei "huge" permit adresarea întregului spaţiu de memorie. Diferenţa între ele este determinată de mărimea obiectelor de administrat. În timp ce pointerii "far adresează obiecte nu mai mari de 64 kbyte, cu pointeri "huge" se poate lucra cu obiecte de orice mărime. De aceea, la incrementarea respectiv decrementarea unui pointer "huge" se foloseşte o rutină mai complexă, care să controleze depăşirile de segment. Aceste diferenţe sînt demonstrate de programul care urmează. În timp ce pointerul "far" nu "se prinde" la ieşirea din limitele segmentului, pointerul "huge" modifică valoarea de segment.

(Computer Persönlich, 8/91)

Outsourcing

Păcatele scumpe ale conducătorilor oficiilor de calcul

Preşedintele avea o bănuială: oficiul de calcul lucrează ineficient. De-abia comanda către o întreprindere externă pentru preluarea prelucrării datelor ("outsourcing") a relevat potenţialul de scădera a costurilor. Outsourcing este în multe cazuri calea corectă pentru a evidenţia scumpele păcate ale conducătorilor oficiilor de calcul şi de a organiza prelucrarea datelor mai eficient.

Conducătorul oficiului de calcul era foarte calificat. 20 de ani de vechime în meserie. Consilierii de la Programator au ştiut imediat ce este de făcut: i-au dat un alt job, în care nu mai avea nici o legătură cu patronul lui de pînă acum. Iar oficiul de calcul (secţia de prelucrare a datelor) al ramurii din Germania a concernului scandinav Nokia a fost preluat de Programator, de personal care nu a avut pînă atunci nimic de-a face cu Nokia.

Programator procedează întotdeauna ca la Nokia. Întreprinderea scandinavă este specializată în a prelua în regie proprie oficiile de calcul ale întreprinderilor mari (cu peste 20 de angajaţi). Programator promite o scădere drastică a costurilor prelucrării datelor după preluare. Conform experienţei scandinavilor, cu personal extern pot fi evitate dezvoltări greşite în oficiile de calcul ale întreprinderilor.

Strategia de afaceri proprie Programator se hrăneşte din erori pe care oficiile de calcul "ale casei" le fac aproape întotdeauna în întreprinderile mari. Catastiful păcatelor şefilor oficiilor de calcul:

- hard se cumpără după buget. Sumele aprobate de preşedinte se cheltuiesc fără discernămînt, chiar dacă nu este necesar. Se cumpără cele mai performante calculatoare care "intră" în buget.

- decizii odată luate sînt apărate cu argumentul "bazei deja instalate". Domină justificarea investiţiilor anterioare. Gînduri asupra unor soluţii mai bune apărute între timp rămîn neformulate.

- Importanţa şefului oficiului de calcul în întreprindere este măsurată atît de către acesta cît şi de conducerea întreprinderii după numărul de persoane din subordine. Impulsurile pentru a organiza preluarea datelor mai eficient sau de a renunţa la lucruri inutile lipsesc.

Programator crede că are reţeta potrivită pentru a combate toate aceste neajunsuri: prelucrarea datelor ca o prestare de serviciu orientată spre piaţă. La client se face o prelevare exactă a datelor de care are nevoie şi cu ce mijloace pot fi ele culese cel mai avantajos. Prestarea realizată de Programator se facturează şi face posibilă, pentru conducerea întreprinderii, evidenţierea a ce servicii pentru cîţi bani s-a făcut.

Cu acest concept, Programator vrea să devină şi în Germania o casă de soft de frunte. În Suedia, întrepinderea s-a afirmat deja ca prestator de servicii informatice. Cu 2500 de angajaţi, Programator este cel mai mare consilier de întreprindere şi de prelucrarea datelor din Scandinavia. În concernul înfiinţar în 1964, de care aparţine şi firma de consultanţă pentru construcţii Jacobsen & Widmark, lucrează circa 4600 de persoane.

De faptul că prelucrarea datelor este mai ieftină ca soluţie conformă cu piaţa decît printr-o secţie proprie s-au lăsat convinse pînă acum printre altele societatea de stat pentru telefoane, Svenska Televerkert, societatea de asigurări Skandia, concernul electrotehnic Ericsson precum şi filiala-mamă a concernului Nokia din Finlanda. Ideea s-a născut în 1979. Atunci Programator a preluat prelucrarea datelor pentru un producător de oţeluri superioare din Karlstad/Suedia. Între timp peste 20 de concerne se numără printre clienţi, de la uzina de ţesut celular pînă la specialistul în tehnica frigului Frigoscandia.

Drumul spre succes a trecut şi prin experienţe dureroase. "De la clienţi noi vrem să preluăm prelucrarea datelor complet, atît ca personal cît şi ca şi capital", declară Hans Peter Clieves, directorul Programator/ Germania. Rigoarea aceasta vine din experienţe nefaste făcute cu joint-ventures (societăţi mixte).

În faza de constituire, în unele cazuri Programator a acceptat participări de 50%. Jumătate din acţiuni aparţineau întreprinderii din care se separa prelucrarea datelor, cealaltă jumătate aparţinea concernului Programator. Modelul nu s-a dovedit viabil. Clieves: "O gîndire consecventă pentru prestarea de servicii nu are nici o şansă, dacă cel care a făcut comanda stă cu 50% în consiliul de conducere."

(I.F.)

8514/A	Advanced Display Adapter
A/D	Analog / Digital
AES	Application Environment System (Atari)
ALU	Arithmetic Logic Unit
ANSI	American National Standards Institute
API	Application Programm Interface
ARLL	Advanced Run Length Limited
ASCII	American Standard Code for Information Interchange
AT	Advanced Technology
ATF	Automatic Track Finding
BIOS	BASIC I/O System
CAD	Computer Aided Design Computer Aided Drafting
CADD	Computer Aided Drafting Design
CAM	Computer Aided Manufacturing
CASE	Computer Aided Systems Engineering
CBT	Computer Based Training
CCD	Charge Coupled Device
CD	Compact Disc
CD-ROM	Compact Disc - Read Only Memory
CDI	Compact Disc Interactive
CGA	Color Graphics Adapter
CISC	Complete Instruction Set Computer
CLI	Command Line Interface
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processor Unit
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DAM	Direct Access Method
DAT	Digital Audio Tape
DBA	Data Base Administrator
DBMS	Data Base Management System
DD	Double Density
DDE	Dinamic Data Exchange
DDL	Data Definition Language
DDS	Digital Data Storage
DEC	Digital Equipment Corporation
DIP	Digital Image Processing
DMA	Direct Memory Access
DML	Data Manipulation Language
DOS	Disk Operating System
DR DOS	Digital Research DOS
DTP	DeskTop Publishing
DVI	Digital Video Interactive
EGA	Enhanced Graphics Adapter
EMM	Expanded Memory Manager
EMS	Expanded Memory Standard
EPROM	Erasable Programmable Read-Only Memory
FAT	File Allocation Table
FBAS	Farb-/Bild-/Austast-/Synchronsignal
GEM	Graphics Environment Manager
HC	Home Computer
HD	High Density
HGC	Hercules Graphics Adapter
HP	Hewlett Packard
HPGL	Hewlett-Packard Graphics Language
IBM	International Busines Machine
IEEE	Institute of Electrical and Electronics Engineers
IFF	Interchange File Format
ISO	International Standards Organization
I/O	Input/Output
JESSI	Joint European Submicron Sillicon
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LIM	Lotus Intel Microsoft
MCA	Micro Channel Architecture
MCGA	Multi Color Graphics Array Memory Controller Gate Array
MDA	Monochrome Display Adapter
MFM	Modified Frequency Modulation
MGA	(Hercules) Monochrome Graphics Adapter
MIDI	Musical Instrument Digital Interface
MIPS	Millions Instructions Per Seconds
MFLOPS	Millions FLOating Points OPeration per Seconds
MMU	Memory Management Unit
MS-DOS	Microsoft DOS
NEAT	New Enhanced Advanced Technology
NVR	Non Volatile RAM
OCR	Optical Character Recognition
OOP	Object Oriented Programming
OS/2	Operating System /2
PAL	Paradox Application Language
PbE	Prompt by Example
PC	Personal Computer
PCI	Personal Computer Instrumentation
PGA	Professional Graphics Adapter
PS	Personal System
QbE	Query by Example
RAM	Random Access Memory
RAW	Read After Write
RGB	Red Green Blue
RISC	Reduced Instruction Set Computer
RLL	Run Length Limited
ROM	Read Only Memory
SAA	Systems Applications Architectur
SCSI	Small Computers System Interface
SDK	System Developement Kit
SIMM	Single In line Memory Module
SQL	Structured Query Language
TFT	Thin Film Transistor
TIFF	Tag Image File Format
TOS	Tramiels Operating System (Atari)
TSR	Termine and Stay Resident
TTL	Transistor-Transistor Logic
VDI	Virtuell Device Interface
VGA	Video Graphics Array
VMS	Virtual Memory System
WYSIWYG	What You See Is What You Get
XGA	eXtended Graphics Array
XT	eXtended Technology

Din scrisorile primite la redacţie

"...la CAS' 85 s-a discutat şi despre faptul că n-avem timp să aşteptăm 20-30 de ani, pînă cînd se recunoaşte că neologismele respective s-au încetăţenit în limba curentă, astfel ca termenii să figureze în DEX şi să nu mai fim acuzaţi de barbarism tehnic.

Dicţionar de specialitate ? Încă din '87 am discutat cu cei de la Editura tehnică (în '88 a fost şi tov. Ilici la CAS) dar spuneau că n-au voie să scoată dicţionare, acesta fiind apanajul EA sau al Editurii ştiinţifice şi enciclopedice. Numai că termenii la care ne referim sînt mai degrabă tehnici, nu ştiinţifici. Poate acum s-au mai schimbat lucrurile.

Cristian Malide, Bucureşti

"Aş dori informaţii asupra posibilităţii de achiziţionare de soft şi documentaţie de firmă la preţuri rezonabile în lei, dacă există furnizori oficiali de soft în România şi care ar fi aceştia.

De asemenea ar fi bine ca revista dvs. să aibă posibilitatea contactării unor eventuali beneficiari ai produselor şi aplicaţiilor noastre (un fel de piaţă de soft). În această întreprindere ar fi interesaţi şi furnizorii de hardware, care fără aplicaţii nu pot exista. Anunţurile noastre nu sînt prea eficiente deoarece revista nu este citită de prea mulţi doritori de aplicaţii ci mai degrabă de specialişti (furnizori). Ori este foarte greu să ne vindem unul altuia programe care întîmplător fac lucruri asemănătoare.

Constatînd cu tristeţe că nici anunţurile din ziarele de mare tiraj nu prea au efect pentru că marea masă a cititorilor nu este interesată în specialitatea noastră, iar directorii de întreprinderi sînt prea ocupaţi ca să citească şi mica publicitate ne vedem în situaţia de a nu putea oferi (chiar licita) programe utile uneori multora.

Aş sugera, dacă s-ar putea, să existe o "bursă" a programelor, prin care conducătorii unităţilor economice să fie atenţionaţi şi interesaţi în achiziţionarea de software. Altfel în curînd vom muri de foame."

Năstase Ileana, Bucureşti

Nu sînt atît de sceptic ca şi corespondenta noastră, dar evident nici mulţumit de ce (nu) am reuşit. Sînt multe cauzele ştiute sau bănuite; o mare parte din ele sînt inevitabile, sînt "datul problemei", dar o altă mare parte sînt vina noastră. Dacă este adevărat că anunţurile de vînzare/cumpărare hard şi cumpărare soft (aplicaţii) sînt de regulă mult prea "strînse" în timp ca să nu devină total neinteresante prin publicarea la un termen incert şi oricum la distanţă mare în timp - lucru pe care încă sperăm că vom reuşi să-l contracarăm, astfel încît revista noastră să ajungă în chioşcuri şi la abonaţi la dată fixă în fiecare lună - oricum lipsa anunţurilor de vînzare soft mă nedumereşte. Dacă am scris o aplicaţie şi am vîndut-o deja odată sau de cîteva ori, ea a ajuns deci la o anumită maturitate şi rotunjime, nu poate fi decît neglijenţă sau dezinteres în a nu repeta cel puţin de cîteva ori un anunţ care să o facă cunoscută în ţară, permiţîndu-mi să o vînd în zeci sau sute de locuri! Investiţia - financiară şi de timp -este derizorie în comparaţie cu cîştigurile probabile. Ideea că revista nu este citită de utilizatorii finali este, după părerea mea, doar parţial adevărată: chiar dacă e clar că nu am să fac reclamă unui program care-l concurează pe al meu, eu voi fi interesat să ştiu cît mai multe despre programe care să-l completeze sau să aducă alte foloase beneficiarilor pe care îi deservesc, şi care vor veni cu atît mai încrezători la mine cu cît li s-a întîmplat mai des să fie mulţumiţi de ceea ce au putut cumpăra de la mine - sau la recomandarea mea. Ca să nu vorbim de faptul că în mod normal pot conveni cu ceilalţi realizatori de aplicaţii pentru a încasa un procent pentru orice vînzare realizată prin intermedierea mea.

Vrînd-nevrînd, va trebui să ne intereseze mult mai mult desfacerea - lucru care pînă acum s-a realizat de regulă atît de defectuos încît nici nu merită pomenit. Pe de altă parte, cred că noi - profesioniştii în informatică - va trebui să facem un pas sau chiar mai mulţi înspre beneficiari, şi nu să aşteptăm să înveţe ei lucrurile pe care noi le ştim, fără ca nouă să ne pese prea mult de ce îi doare pe ei. Dar, mai ales, va trebui să mărim scara la care sîntem obişnuiţi să lucrăm - să nu ne mulţumim să vindem o aplicaţie scump, de cîteva ori, aşteptînd apoi ca beneficiarii să se înghesuie bazat doar pe reclama implicită pe care şi-o face produsul. Cu "bursa", vom mai vedea, vrem, dar n-avem putere să promitem.

Şi, în încheiere, scuze tuturor acelor care ne scriu şi cărora - deşi ar trebui să o facem - nu ajungem să le răspundem în mod individual.

Să auzim (numai) de bine!

Iosif Fettich

Mica publicitate

Ofer programe şi jocuri pentru calculatoare ATARI. Garantez calitatea înregistrărilor. Tel. 985/10.609.

Execut orice lucrări de proiectare de sistem şi aplicaţii, service, cursuri pentru firme posesoare de PC-uri ca şi colaborator extern sau angajat temporar. Tel. 90/42.25.05.

În numărul viitor:

- **La ce e bun codul de bare**
- **EMS - referinţă**
- **Şi totuşi, Cobol?...**

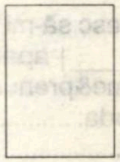
Important !

Revista "**if**" îşi propune editarea unui catalog al întreprinderilor (de stat sau particulare) a căror sferă de preocupări are tangenţă cu informatica (producători hard, producători soft, depanatori, distribuitori, comercianţi, etc.).
Catalogul va conţine adresele întreprinderilor, o scurtă descriere a obiectului de activitate al acestora şi o prezentare succintă a produselor şi serviciilor oferite.
Catalogul va fi distribuit prin reţeaua de difuzare a revistei "**if**", putînd ajunge astfel direct la persoanele interesate. Întreprinderile interesate de publicarea acestui catalog sînt rugate să ia legătura cu redacţia.

Important !

Căutăm difuzori/distribuitori autorizaţi pentru difuzarea revistei "**if**" !.
Condiţii avantajoase !
Rugăm persoanele interesate să ia legătura cu redacţia.

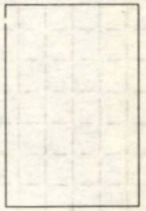
Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !



Micro ATCI

C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !



Micro ATCI

C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if,** pe o perioadă de |_____| apariţii, pe adresa:

Nume&prenume .....

Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....

Cod: ..... Localitatea: ..... Judeţ (sector) .....

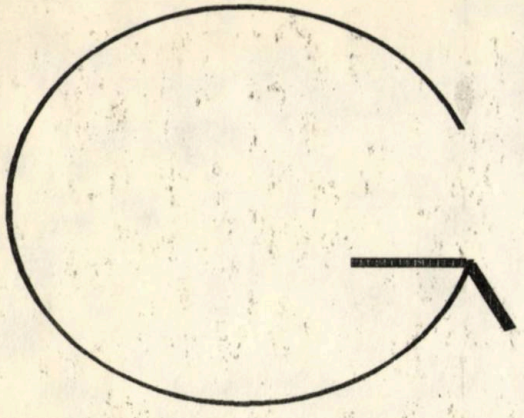
Preţul unui exemplar este 57 lei. Preţul unui abonament se va calcula cu formula: 57 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

If 5 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_



GENESYS

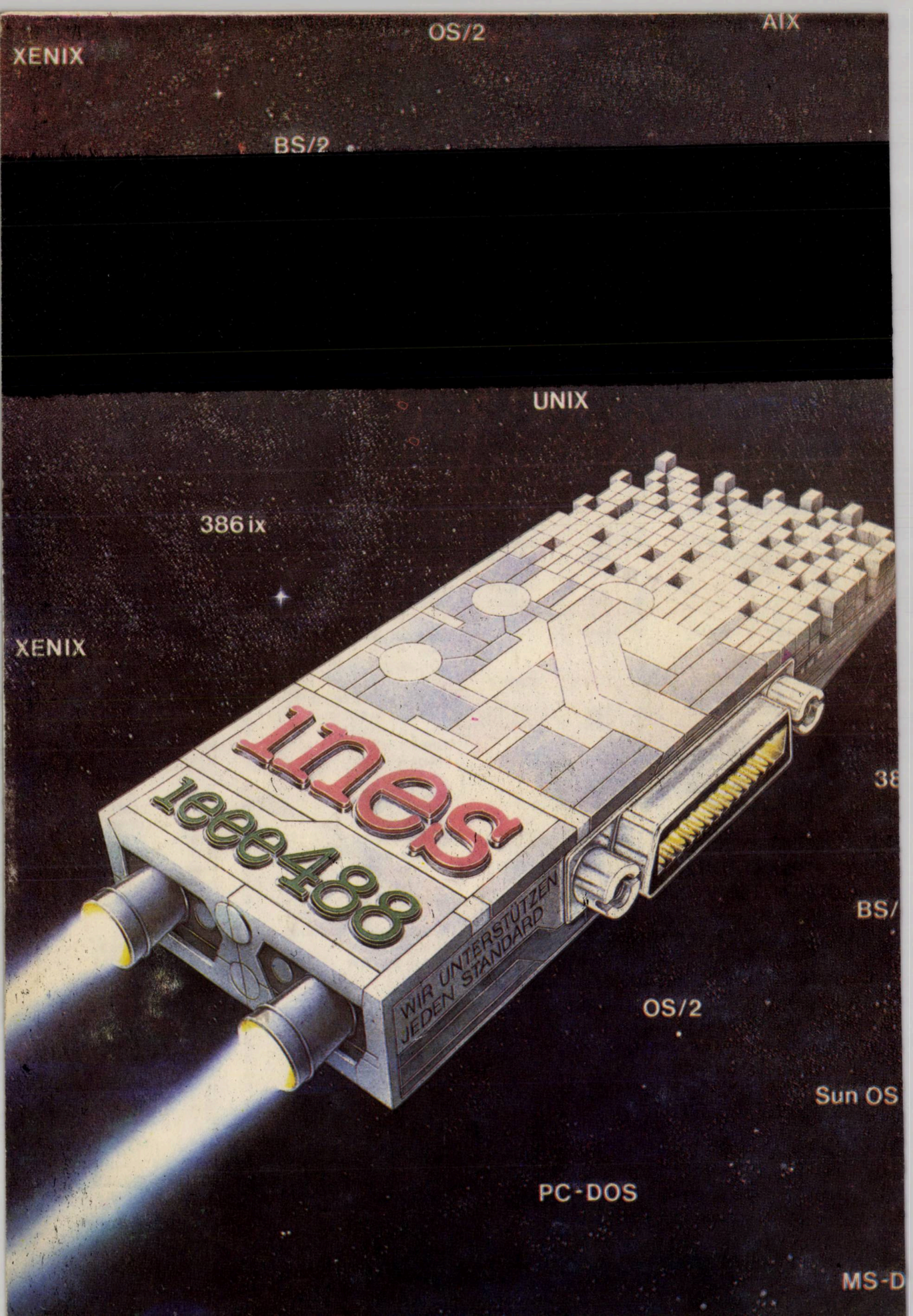
SZINTEZIS COMPUTER GYÖR
UNGARIA

GENESYS ORADEA
ROMANIA

str. Republicii nr. 35
3700 Oradea
tel: 991/31516

Denumire		(N)	Preţ (Valută)	Preţ (Lei)
Calculatoare XT				
GF XTT,,640k,,Mo	$		311	78486.3
GF XT,360k,640k,20M,Mo	$		693	174890.7
4D XTTbill,,640k,,,Arcn.	$		600	151420.5
Calculatoare AT 286				
GF AT-12,,1M,,Mo	$		366	92366.5
GF AT-12,1.2M,1M,40M,Mo	$		938	236720.7
GF AT-12,1.2M,1M,80M,Mo	$		1366	344734.0
GF AT-16,,1M,,Mo	$		393	99180.4
GF AT-16,1.2M,1M,80M,Mo	$		1394	351800.3
OFF AT-16,,1M,,Mo,Arcn.	$		948	239244.4
AT-16N,,1M,,Mo	$		518	130726.4
AT-16N,1.2M,1M,40M,Mo	$		1090	275080.6
terminale DTK				
DTK AT-16,1M,1.2,40M,Mo	A		10440	223063.6
DTK 386SX,1M,1.2Mf,noMO - 16Mhz,VGA,D-Top,noKey	A		9132	195116.6
- 20 Mhz,VGA,D-Top,noKey	A		10361	221375.7
DTK486-33M w 64K C.TOW - 4M,noW,FD,VGA,noMo,noKey	A		45077	963126.4
Calculatoare AT-386				
AT386,mainb,33,0M,nC	$		842	212493.4
386,12/25,1.2M,1M,40M,Mo	$		1526	385112.8
386,20/25,1.2M,2M,80M,Mo	$		2139	539814.1
CH.LT3600 16/20,40W,VLCD	$		3036	766187.7
LT.LA30,AT,1M,20W,MLCD	$		1960	494640.3
Monitoare				
12" mono FUJIT,VALTR	';U8.1			
14" mono SUPERTRON	$		110	27760.4
14" EGA SUPERTRON	$		407	102713.6
14" VGA (1024x768) SUPERT	$		414	104480.1
20" VGA monitor+cartela	A		86400	1846043.9
15" MAG allo A4	$		869	219307.4
Adaptoare grafice				
cartela mono	$		25	6309.2
cartela EGA	$		81	20441.8
VGA,8b,256k(800x600)	$		81	20441.8
VGA,16b,512k(1024x768)	$		138	34826.7
Subansamble				
FD 1.2Mb	$		81	20441.8
FD 1.44Mb	$		81	20441.8
FD contr. 2 driv. AT	$		25	6309.2
WD 1003 contr.	$		62	15646.8
controller SCSI	$		138	34826.7
Keyboard 101	$		46	11608.9

Denumire		(N)	Preţ (Valută)	Preţ (Lei)
GM6 mouse	$		21	5299.7
MICROSCAN	$		186	46940.4
COPROCESOR 287-8	$		168	42397.7
COPROCESOR 387-33	$		904	228140.2
MIDI tower+sursa	A		1747	37326.8
Discuri Winchester				
20 MB st225 (65ms)	$		233	58801.6
40 MB st251-1 (28ms)	$		331	83533.6
80 MB st4096 (28ms)	$		669	168833.9
1.2 GB CDC Winc.	$		5505	1389283.1
controller SCSI	$		138	34826.7
Memorii şi accesorii				
1 MB	$		66	16656.3
IC 41256-08	$		7	1766.6
Elemente de reţea				
ARCNET 8bit STAR-BUS	$		61	15394.4
Ethernet NE1000 8bit	$		138	34826.7
Etherlink 3COM 8bit	$		304	76719.7
Novell SFT 2.15	$		5654	1426885.8
Imprimante				
EPSON LX 800	D		235	35301.4
EPSON FX 1050	D		852	127986.4
EPSON LQ 2550	D		2760	414603.9
HP LaserJet III	$		2553	644294.2
HP LaserJet II/P	$		1604	404797.5
STAR NX1001	$		226	57035.1
Cartridge FX1050	$		5	1261.8
Surse neintreruptibile				
UPS 550 W	$		311	78486.3
APC UPS 1200 W	$		1228	309907.3
Altele				
Streamer Wangtech 60 MB IN	$		704	177666.7
Modem INT 1200/SM12H	$		94	23722.5
3.5" DS/DD floppy 3M	$		32	8075.8
3.5" DS/HD floppy 3M	$		40	10094.7
5.25" DS/DD floppy 3M	$		9	2271.3
5.25" DS/HD floppy 3M	$		16	4037.9
Leporello 240x1x12x1700	$		16	4037.9
Leporello 382x2x12x900	$		53	13375.5
Telefoane				
PHIL. D9026 (10 mem)	$		54	13640
CANON FAX 80	D		973	146162.9
SHARP 7850	$		2835	715461.9


OS/2
AIX
XENIX
BS/2
UNIX
386 ix
XENIX
ines
ieee488
WIR UNTERSTÜTZEN
JEDEN STANDARD
OS/2
Sun OS
PC-DOS

CALCULATOARE
PERSONALE

Revistă lunară editată de Micro ATCI S.R.L. Tîrgu Mureş

6 / 91
(9)
57 Lei

Staţiile de lucru Next

Memento:
Specificaţiile XMS

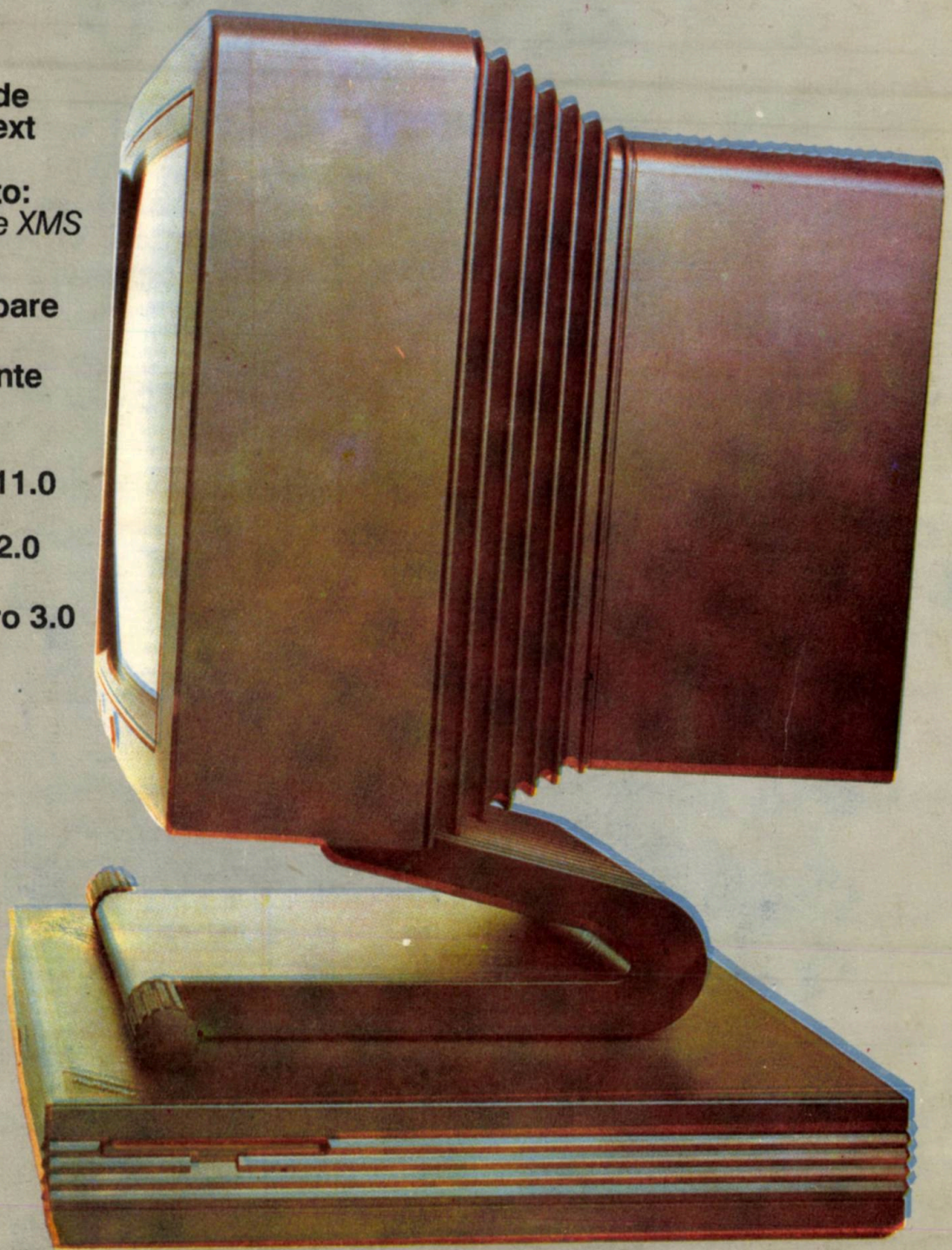
Cod de bare

Imprimante laser

Autocad 11.0

Foxpro 2.0

Quattro Pro 3.0



o NOU o NOU o NOU o NOU o NOU o NOU o NOU o NOU o

actiune E G A la

AbMod !

SISTEME COMPATIBILE IBM/PC la numai

239.000 lei

AT 286 12 MHz, 1 MB RAM, HD 40 MB, 1 unitate floppy 1.2 MB 5.25", 1 unitate floppy 1.44 MB 3.5", Monitor color EGA, 1 Mouse, 1 set dischete (10 buc) 5.25", 1 set dischete (10 buc) 3.5", 1 interfata seriala, 1 interfata paralela, tastatura 101 taste

ATENTIE ! Pentru comenzi en-gros firma acorda reduceri substantiale.

Pentru alte echipamente de **calcul** si **birotica**, solicitati lista noastra completa de preturi.

GARANTIE 1 AN DE LA LIVRARE !

ASTEPTAM COMENZI FERME LA UNA DIN ADRESELE:

3400 CLUJ-NAPOCA MAGAZIN AbMod
B-dul 22 Decembrie Nr.135 Tel: 95/156350
3400 CLUJ-NAPOCA Sediul central AbMod
Str. Motilor Nr.11 Tel, Fax: 95/111090 Telex: 31445
3700 ORADEA Str. Dacia Nr.40
Tel: 991/60278 991/44476 Fax: 991/56881

NU UITATI ! Stocul este limitat, si oferta este valabila pina la epuizarea stocului

o NOU o NOU o NOU o NOU o NOU o NOU o NOU o NOU o

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:

ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:

ing. Ioan Cozac
ing. Alin Flaidăr
ing. Alin Udrea

Tiparul: tipografia Tîrgu Mureş
Tipografi:
Halaţiu Mihai,
Cormoş Mircea

Revista apare lunar.

Preţ: 57 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Adresa şi telefonul redacţiei:
Micro ATCI,
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel. 954/31660 (direct),
33.612, 24.158, 20.057, 33.511,
int. 134 sau 189
41417 sau 46885 (după ora 19)
fax 954/35208,
telex 65354.

Pictures at an Exhibition

Am fost şi noi la TIBCO '91. Am fost, am văzut şi ne-am întors acasă. Am văzut, am întrebat, am ascultat, am simţit. Căldură mare. Înghesuială. Bunuri de consum. Deocamdată mai mult pe la tîrguri. Cine ştie, poate ... Nu îndrăznesc să-mi duc gîndul mai departe. Simt că devin mai uşor. Vîntul liberalizării îmi umflă buzunarele. Dar să revin ... Ce ne interesează pe noi ? Pe noi ne interesează calculatoarele personale. La TIBCO '91 am numărat cel puţin 30 de întreprinderi care se ocupau (şi) cu problema în cauză.

Pozitiv este faptul că se poate cumpăra aproape orice: calculatoare, periferice, soft-uri, materiale consumabile. Bani să fie ! Dar apropo de bani, este de remarcat faptul că în timp ce preţurile la mai toate produsele cresc după curbe exponenţiale, la tehnica de calcul scad. Cu toate acestea preţurile practicate sînt accesibile doar întreprinderilor. Mai aşteptăm. Sper să nu-mi creez antipatii afirmînd că expoziţia mi-a evocat un "Ballet of the unhatched chickens". După cum ne sfătuieşte institutul pentru prospectare a pieţii Frost & Sullivan din Marea Britanie, (vezi articolul "O piaţă dificilă !"), bobocii va trebui să-i numărăm abia în toamna lui 1993. Sîntem optimişti. În loc să ne supărăm pentru faptul că ne situăm pe ultima poziţie în grafic, ne bucurăm că figurăm. Albania nu apare ! Sperăm, deci, că în 1993 în România se va realiza în tehnica de calcul şi informatică o cifră de afaceri de 135,63 milioane de dolari. Şi mai mult decît atît !

Tot pentru că sîntem optimişti vă propunem spre lectură şi articolul despre codurile de bare. Articolele despre limbajul de programare Cobol, vă vor determina, poate, să renunţaţi la asocierea: Cobol - preistorie, iar prezentarea celor mai noi versiuni Foxpro, Quattro şi Autocad vă va da, sperăm, o imagine a ameţitorului ritm în care se trăieşte astăzi în lumea informaticii.

Romulus Maier

MAI MULT TACT!
SUS
VREM RAMA!
FARĂ VIRUȘI!
JOS
IMPLEMENTA-NE-ATI!

Cuprins

Ştiri

- PC Tools al VII-lea pag.4
- Upper Memory pentru DOS 5.0 . . pag.4
- Basic sub Windows pag.4
- O piaţă dificilă ! pag.5
- Oglinda preţurilor (mai 1991) pag.5

Hardware

486SX - Putere pentru toată lumea . . pag.6

IBM în frunte pag.7

Doar la o zi de la anunţarea oficială a procesorului Intel 486SX, IBM a şi dat o soluţie pentru modelele 90 şi 95. În plus a propus un viitor produs posibil cu o cartelă 486/50 MHz pentru cele două modele.

Vă prezentăm

Next please pag.8

Quattro Pro 3.0 . . . . . pag.10

CAD

Computer Aided Design pag.12

Test: Auto - CAD 11.0 . . . . . pag.14

Versiunea 11 a clasicului "Auto-CAD" a fost anunţată încă în urmă cu un an. Acum este disponibilă, în sfîrşit, prima versiune, fapt care ne prilejuieşte prezentarea unui prim test al produsului.

Aplicaţii

Cod de bare pag.16

Imprimante laser

Cum funcţionează imprimantele laser? pag.24

Sinteză: Imprimante laser la preţuri sub 5.000 DM pag.26

Imprimanta HP Laserjet IIIP pag.27

Software

Specificatia Lotus/Intel/Microsoft/Ast (XMS) pentru memoria extinsa . . . . supliment

Cobol = Cobol ? pag.33

MS - Cobol V 4.0 PDS pag.36

Cel ce-şi aduce aminte de primul compilator Cobol pentru PC va rămîne perplex în faţa celei mai noi versiuni. Editoarele confortabile şi multele programe utile uşurează foarte mult munca programatorului profesional. În sfîrşit te poţi concentra asupra problemei principale - programarea. Multe munci suplimentare care înainte cădeau în sarcina programatorului sînt preluate acum de compilator.

Foxpro 2.0 . . . . . pag.38

În cursa bazelor de date, Foxpro 2.0 a luat din nou un avans în faţa concurenţilor dBase IV şi Paradox 3.5.

Cursuri

Reţele neuronale partea a IV-a pag.40

În ultimii ani, recunoaşterea scrisului prin calculator s-a perfecţionat mereu. Recunoaşterea de caractere prin intermediul reţelelor neuronale este un pas într-o direcţie promiţătoare.

OOP pag.44

Practică

- Protecţie prin parolă pag.50
- Vă place muzica ? pag.52
- Ştergere rapidă pag.52
- ChDir confortabil pag.53

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Editorial . . . . . pag. 1
- Poşta redacţiei pag.54
- Mica publicitate pag.54

PC Tools al VII-lea

Versiunea 7.0 a lui PC Tools a început o nouă rundă în confruntarea pentru titlul de cel mai bun program utilitar. Noile atu-uri: suport reţea şi suport Windows.

Săptămînile trecute, Central Point Software, din SUA, a anunţat a şaptea versiune a programului utilitar PC Tools. Noua versiune conţine, după declaraţiile lui Corey Smith, preşedintele CPS, 80 de detalii îmbunătăţite şi 10 module noi, care, ca de obicei, pot fi apelate de sub o suprafaţă utilizator unitară. PC Tools devine primul program utilitar care poate fi rulat atît sub MS-DOS cît şi sub Windows şi a cărui mod de operare este identic în ambele medii.

Printre cele mai importante noutăţi se numără un program de transfer de date, asemănător cu Laplink, cu ajutorul căruia, prin intermediul unui modem, a unei reţele, sau a unui cablu, poate fi comandat un calculator de la distanţă. Transferul de date între cele două calculatoare este posibil cu ajutorul programului Commute. Posibilitatea lucrului în reţea joacă un rol important atît în cazul lui Commute cît şi în cazul celorlalte module. Astfel "Undelete" poate recupera acum şi fişiere aflate pe un server Novell, iar Filefind utilizează atributele extinse de reţea. Printre celelalte noutăţi se numără un program antivirus şi alte utilitare cunoscute deja din programele concurenţei, cum ar fi "Wipe" care şterge un fişier astfel încît să nu mai poată fi recuperat, sau "FileFix" care poate recupera fişiere dBase sau Lotus distruse.

PC Tools costă 179$ în SUA şi trebuia să fie livrat începînd din luna mai.

Basic sub Windows

Visual Basic, produs de Microsoft, permite elaborarea unor aplicaţii orientate obiect sub suprafaţa utilizator grafică Windows.

La un an de la prezentarea lui Windows 3.0, Microsoft prezintă primul limbaj de programare orientat obiect care să lucreze sub această suprafaţă utilizator: Visual Basic. Acest software a fost prezentat la tîrgul de primăvară Comdex din Atlanta, USA.

Analiştii pieţii din SUA estimează că această prezentare este o mişcare strategică importantă pentru Microsoft. David Buyer, analist la Montgomery Securities: "Visual Basic va determina o şi mai mare răspîndire a lui Windows 3.0 în întreprinderi şi va duce la o creştere a cifrei de afaceri Microsoft. Dezvoltarea de aplicaţii Windows este simplificată semnificativ. Visual Basic va deveni fundamentul multor programe Windows". Visual Basic constă, în mare, din două componente centrale: "Forms", pentru cuplarea suprafeţei grafice cu cunoscutele obiecte Windows, şi "Program Units" care conţin procedurile limbajului.

Rick Sherlund, analist la Goldman Sachs, comentează: "Momentan, cu doar 90 milioane dolari, limbajele de programare reprezintă doar o mică parte din cifra de afaceri Microsoft. După lansarea pe piaţă a lui Visual Basic se aşteaptă rate de creştere de 30 pînă la 50% la limbajele de programare.

În SUA, Visual Basic va costa cca. 250$.

Plăci de extensie

Upper Memory pentru DOS 5.0

Întreprinderea Dawicontrol din Göttingen produce o placă de extensie de memorie pentru MS-DOS 5.0, "Upper Memory Board", care poate fi montată în toate PC-urile, de la XT şi pînă la 486. Cea mai nouă versiune a sistemului de operare MS-DOS recunoaşte, în gestionarea memoriei, aşa-numitele Upper Memory Blocks (UPB). Cu ajutorul lor driver-ele şi programele rezidente pot fi încărcate în memoria superioară (Upper Memory). În acest mod utilizatorului îi rămîn disponibili pînă la 600 KByte din memoria de bază.

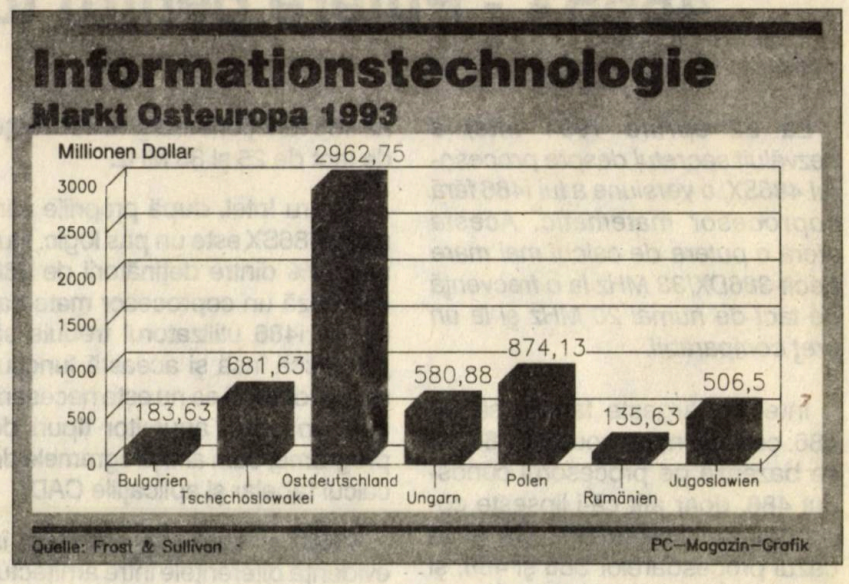
Cartela Upper Memory pune la dispoziţia MS-DOS 5.0 spaţiul fizic RAM necesar. În plus, spaţiul disponibil în extensia de memorie poate fi utilizat, la nevoie, pentru extinderea memoriei de bază. Memoria disponibilă pe placa de extensie, poate fi împărţită în ferestre de cîte 16 KByte care pot fi inserate, la dorinţă, în interiorul spaţiului de adresare cuprins între 512 KByte şi 1 MByte.

Programul de instalare stabileşte domeniile RAM libere din sistem şi preia, orientat dialog, întreaga configurare. Reglarea prin comutatoare DIP sau jumperi a fost înlăturată în întregime. În configuraţia de bază, placa este echipată cu 256 KByte şi costă 295 DM. La cerere ea poate fi echipată cu pînă la 512 KByte RAM.

Informatii: Dawicontrol. Leinestr. 36, 3400 Göttingen, Tel. 0551/72075.

O piaţă dificilă !

Producătorii de calculatoare care doresc să fie prezenţi în Europa de Est trebuie să aibă nervii tari. Ei trebuie să investească încă de pe acum, dar nu se pot aştepta la cîştiguri decît după 1993, afirmă institutul de prospectare a pieţii Frost & Sullivan într-un studiu.



Febra aurului şi speranţa într-un cîştig rapid, după introducerea economiei de piaţă în ţările din estul Europei, sînt deplasate, este de părere institutul de prospectare a pieţii Frost & Sullivan, în urma unui studiu despre posibilităţile pieţelor esteuropene. "Această piaţă nu este pentru cei cu nervii slabi", se afirmă în studiu, deoarece şanse au doar aceia care investesc acum pentru a-şi asigura imediat prezenţa pe piaţă. Cîştigul rezultat din aceste investiţii s-ar putea să apară, însă, abia după anul 1993.

Frost & Sullivan a stabilit că pentru anul trecut cifra de afaceri realizată de industria occidentală de tehnică de calcul şi informatică în ţările din estul Europei, încluzînd fostul RDG, se ridică la 1,68 miliarde de dolari. Această cifră ar putea să crească în anii următori la 4,86 miliarde de dolari. Luînd în calcul şi o posibilă rată de eroare se pronostichează că în varianta pesimistă această cifră ar putea urca doar pînă la 3,8 miliarde de dolari, în timp ce în varianta optimistă această cifră de afaceri ar putea atinge în 1993 6 miliarde de dolari (vezi graficul alăturat).

Cea mai mare piaţă esteuropeană se va situa în următorii ani pe teritoriul fostului RDG. Aceasta se explică şi prin faptul că în fostul CAER RDG-ul dispunea de cea mai puternică economie socialistă. Frost & Sullivan: "În fostul RDG, motorul principal al dezvoltării industriale în fostul CAER, se vor desface aproape jumătate din livrările următorilor ani".

Pentru întreprinderile care nu vin din vest, dezvoltarea pieţii se va face mai degrabă lent: "La început cea mai mare cerere va exista pentru calculatoare personale. Ulterior se vor vinde destul de bine şi echipamente Unix şi calculatoare de marcă pentru întreprinderi de dimensiune medie. Numai în landurile fostului RDG va exista încă de la început şi o piaţă semnificativă pentru calculatoare mari."

Ritmul lent, previzibil, de dezvoltare a ţărilor din estul Europei nu trebuie să-i determine pe ofertanţi să tragă concluzia că pot aştepta încă, afirmă Frost & Sullivan. Fundamentul pentru un succes în 1993 trebuie pus încă de pe acum. "Investiţiile necesare sînt semnificative", atrag atenţia cercetătorii londonezi, dar "importantă este o pătrundere rapidă pe piaţă, adesea prin înfiinţarea de firme mixte cu întreprinderile locale." Mai multe întreprinderi, cu o mare putere financiară, au păşit deja pe acest drum, printre altele IBM, Siemens şi DEC. În ciuda timpilor mari de aşteptare, aceste întreprinderi au recunoscut: "Aceia care vor pătrunde cel mai rapid pe aceste pieţe vor avea în viitor cele mai bune perspective".

(R.M.)

Oglinda preţurilor (mai 1991)

Sisteme	Preţ de stradă (no-name)	Aparate de marcă ca de ex.
AT 80286-CPU, 1 MByte RAM harddisk de 40 MByte	1.248 DM	Compaq 286 N1/SL 3.780 DM
386SX 80386SX-CPU, 1 MByte RAM, harddisk de 40 MByte, o unitate floppy, monitor VGA	2.490 DM	Compaq 386 N1/SL 4.415 DM
386/33 80386 CPU, 2 MByte RAM, harddisk de 180 MByte, o unitate floppy, monitor VGA	4.408 DM	IBM-PS/2-80 A-31 12.350 DM
486/33 80486-CPU, 4 MByte RAM, harddisk de 330 MByte, o unitate floppy, monitor VGA	9.450 DM	Compaq 486 23.695 DM

486SX - Putere pentru toată lumea

La 22 aprilie 1991 Intel a dezvăluit secretul despre procesorul 486SX, o versiune a lui i486 fără coprocesor matematic. Acesta oferă o putere de calcul mai mare decît 386DX/33 MHz la o frecvenţă de tact de numai 20 MHz şi la un preţ comparabil.

Intel îşi lărgeşte familia sa de 486: cea mai mare noutate, 486SX, se bazează pe procesorul cunoscut 486, doar atît că-i lipseşte coprocesorul matematic. Ca şi în cazul procesoarelor 386 şi 486, şi în acest caz este vorba despre un procesor pur de 32 de biţi, compatibil integral soft cu predecesorii săi din familia Intel x86.

Performanţele de calcul ale noului venit, date de Intel sînt de 16.5 MIPS (milioane de instrucţiuni pe secundă), la o frecvenţă de tact maximă de 20 MHz. Comparativ 386DX ajunge la 11.3 MIPS, la o frecvenţă de tact de 33 MHz. În ciuda saltului calitativ clar, procesorul 486SX va costa doar cu 20 pînă la 30 DM mai mult decît un 386DX. David House, de la Intel, comentează intrarea pe piaţă astfel: "Procesorul Intel 486SX deschide utilizatorului drumul spre calculatoarele care-şi merită banii, posedînd performanţele superioare ale tehnologiei 486".

Cu 486SX, specialistul în procesoare copiază strategia de succes a procesorului 386SX. După produsul inovator 486 apare deci o variantă mai redusă, dar mai accesibilă ca preţ.

Pentru 486SX, Intel oferă şi coprocesorul matematic 487SX, care face calculatoarele SX comparabile cu versiunea completă 486. Combinaţia 486SX - 487SX costă totuşi mai mult decît un 486. O altă limitare este faptul că 486SX se produce deocamdată numai în varianta 20 MHz, în timp ce procesorul 486 funcţionează şi la frecvenţe de tact de 25 şi 33 MHz.

Pentru Intel, după propriile afirmaţii, 486SX este un pas logic. Numai 30% dintre deţinătorii de 386 utilizează un coprocesor matematic. La i486 utilizatorul trebuie să plătească însă şi această funcţiune, cu toate că ea nu este necesară decît în cazul anumitor tipuri de programe, cum ar fi programele de calcul tabelar şi aplicaţiile CAD.

486SX scoate mai pregnant în evidenţă diferenţele între arhitecturile lui 386 şi 486, deoarece i486 este mai mult decît "doar" un procesor 386 cu cache şi coprocesor. Ultimele modele ale familiei x86 au fost îmbunătăţite ca design faţă de 386, ceea ce explică şi avantajul vitezei sporite în cazul lui 486SX.

După datele furnizate de Intel, arhitectura 486, la acelaşi tact, permite o putere de două pînă la trei ori mai mare decît cea a procesoarelor 386. Cu toate că familia x86 aparţine familiei procesoarelor CISC (Complete Instruction Set Computer), 486 şi 486SX prelucrează multe comenzi ca un procesor RISC (Reduced Instruction Set Computer) în cadrul unui ciclu de tact.

Un avantaj hotărîtor vis à vis de arhitectura 386 este timingul modificat al procesorului: i486 utilizează fiecare impuls al cuarţului, în timp ce 386 necesită cuarţuri de două ori mai rapide pentru a putea atinge aceeaşi rată de tact efectivă. În toate procesoarele 386 la 33 MHz bate deci inima unui cuarţ de 66 MHz, ceea ce implică măsuri speciale de ecranare. Din această cauză crearea unor procesoare 386 cu o frecvenţă de tact mai ridicată devine o problemă, deoarece un cuarţ de 80 sau de 100 MHz devine tot mai greu de ecranat. Arhitectura 486 foloseşte mai economicos ciclurile de tact şi din acest motiv poate lucra mai uşor la un tact de 50 MHz sau chiar de 100 MHz. Procesorul 486 la 50 MHz nu a fost încă anunţat oficial, dar diverşi producători de calculatoare, printre care şi IBM, vorbesc deja despre el, şi îşi planifică să-l folosească cît mai rapid posibil.

Tactul de 20 MHz mai aduce un avantaj lui 486SX: producătorii de PC-uri pot adăuga componente periferice convenabile, deoarece o placă de bază 486SX, per total, ar trebui să fie mai ieftină decît o placă de bază 386DX. Viteza procesorului 486SX este influenţată mai ales de cache-ul integrat, care lipseşte la 386.

În ciuda tuturor avantajelor, 486SX le va spori clienţilor chinul alegerii: să te decizi pentru un 386, sau mai bine să aştepţi un 486SX ? Merită să-ţi cumperi un 486 sau pentru anii următori ajunge şi un 486SX ? Răspunsurile la aceste întrebări vor fi date în lunile următoare de către producătorii de PC-uri. De preţurile pe care le vor practica va depinde cît de repede se va impune 486SX.

Specialiştii apreciază că vînzările de 386 vor scădea sensibil, dacă producătorii vor transfera avantajul preţului mai scăzut şi asupra produselor finale. Dacă însă vor pretinde un preţ suplimentar pentru puterea sporită, atunci 386 mai are încă şanse bune de a se menţine pe piaţă. Cît de rapid va reuşi deci 486SX să-l înlocuiască pe 386 pe piaţă rămîne de văzut.

Politica de preţuri agresivă, practicată de Intel, lasă să se întrevadă faptul că în Santa Clara schimbarea este dorită mai degrabă astăzi decît mîine. Motivul ar putea fi "copierea" lui 386 de către AMD, căruia 486SX i-ar putea lua piuitul.

(I. M. & R. M.)

IBM este în frunte

Doar la o zi de la anunţarea oficială a procesorului Intel 486SX, IBM a şi dat o soluţie pentru modelele 90 şi 95. În plus a propus un viitor produs posibil cu o cartelă 486/50 MHz pentru cele două modele.

IBM vrea să fie din nou în fruntea dezvoltării tehnice, acest lucru l-a demonstrat Big Blue (Marele Albastru) la 23 aprilie în München. Doar la o zi după ce Intel şi-a prezentat noul procesor 486SX IBM şi-a prezentat modelele PS/2 90 şi 95 dotate cu acest procesor.

Sistemele introduse în octombrie anul trecut folosesc sisteme complexe de procesoare interschimbabile, astfel încît clientul să-şi poată completa mai tîrziu calculatorul cu un 486 de 25 sau de 33 MHz. "Vrem să protejăm investiţia clienţilor noştri" afirmă Dr. Wolfram Ischebeck directorul comercial de la IBM. Clientul nostru va trebui doar să înlocuiască modelul CPU în loc să-şi cumpere alt calculator de capacitate mai mare.

IBM este destul de sigur pe sine cînd afirmă că aceste modele PS/2 nu vor îmbătrîni în următorii 5-7 ani, datorită posibilităţii de actualizare (upgrade) pe care le au, cu toate că pentru 1992 se aşteaptă de la Intel procesoare 586.

Cele 2 variante noi reprezintă configuraţiile iniţiale ale modelelor PS/2 90 şi 95, pe care IBM le-a oferit pînă în prezent doar cu versiunea completă a procesorului 486.

Vedere de ansamblu	
Calculator	**PS/2, Modell 95 XP**
Producător	IBM
Preţ	încă necunoscut
Procesor	486SX
Frecvenţa de tact	20 MHz
Memoria principală	8 MByte
Adaptor grafic	XGA
Harddisk	160 MByte
Controller	SCSI
Sistem magistrală	Microcanal
Cuploare (slot-uri) libere	6

Modelul 90 este un calculator desktop, avînd 3 cuploare de microcanal libere. El este dotat cu o memorie de 4 MByte (care poate fi completată pînă la 32 MByte) şi o placă grafică modernă XGA, care oferă o rezoluţie de 1034x768 pixeli la 256 de culori, sau de 640x400 pixeli la 300.000 culori (16 biţi).

Modelul 95 se aseamănă în mare măsură cu modelul 90 în ceea ce priveşte datele tehnice, doar că dispune de o carcasă tower mult mai mare şi mai robustă. Calculatorul are aproape 22 kg, dispune de pînă la 8 cuploare de microcanal (două din cele 8 sînt deja ocupate de placa grafică şi de controller-ul de harddisk) şi dispune de spaţiu pentru inserarea a încă 7 plăci, de exemplu alte 4 harddiskuri SCSI, care pot mări capacitatea de memorare pînă la peste 2 GByte. Tower-ul se pretează din acest motiv ca server de capacitate mare. Tocmai aici oferta de actualizare (upgrade) a lui IBM devine interesantă, deoarece capacitatea serverului poate fi mărită doar prin cîteva operaţiuni, fără ca să fie necesară schimbarea calculatorului.

În legătură cu strategia de dezvoltare a IBM-ului, s-a prezentat o placă 486 de 50 MHz pentru modelele 90 şi 95. Peter Richard, susţine că ar fi o demonstraţie de tehnologie şi nu de un produs terminat. Placa de 50 MHz ar putea fi în scurt timp o extensie a modelelor 90 şi 95.

Conform afirmaţiilor IBM primele aparate cu procesorul 486SX vor fi livrate în iunie 1991. Preţul nu este încă stabilit.

(I.M.)

Familia x86 dintr-o privire			
Date comparative			
Procesor	**I386**	**I486SX**	**I486**
Anul apariţiei	1986	1991	1989
Frevenţa de tact maximă	33	20	33 (50)
Puterea (MIPS)	11,4	16,5	27

Next please

Acum există calculatoare Ferrari doar la jumătate de preţ: Next, noua firmă a vizionarului care a creat calculatorul Macintosh, Steve Jobs, vrea să cucerească piaţa mondială cu staţii de lucru (Workstation) de 11.000 DM.

Ele sînt domnitoarele reţelelor, maestrele simulărilor complexe, stăpînele graficii color - cu staţiile de lucru începe lumea de dincolo de PC-uri.

Din exterior staţiile de lucru arată ca nişte calculatoare uzuale, inofensive, de birou, dar afară-i vopsit gardul şi înăuntru .... Microprocesoare speciale se întrec în a-i asigura performanţe maxime, ceea ce este bine pentru lucrul în reţea, este bine pentru baze de date mari, bine pentru aplicaţii grafice şi color, bine pentru simulări, dar rău pentru pungă: staţiile de lucru uzuale ale firmelor Hewlett-Packard, Sun sau IBM costă toate peste 20.000 DM.

Această limită a preţurilor se va clătina în curînd. Steve Jobs, excentricul vizionar al lui Apple, de la care a fost nevoit ulterior să plece, intră din nou în arenă. Noua sa firmă Next Computer a adus pe piaţă trei noi staţii de lucru la preţuri tentante: "Nextstation" (cca. 10.800 DM), "Nextstation Color" (cca. 17.300 DM) şi "Nextcube" (cca. 17.300 DM).

Pentru utilizatorii obişnuiţi aceasta este ca şi cum le-ai oferi o maşină Ferrari la jumătate de preţ: senzaţional de ieftin, dar cu toate acestea de neplătit.

Toate cele trei variante sînt echipate cu microprocesoare Motorola 68040 care lucrează cu o frecvenţă de tact de 25 MHz, ceea ce le permite atingerea unei puteri de calcul de 15 MIPS (million instructions per second - milioane de instrucţiuni pe secundă). Pentru comparaţie să amintim că "Macintosh IIci" nu atinge decît 4 MIPS iar calculatoarele MS-DOS echipate cu 386 nu depăşesc nici ele prea mult această valoare.

Memoria principală a calculatorului "Next Station" este de 8 MByte şi poate fi extinsă pînă la 32 MByte.

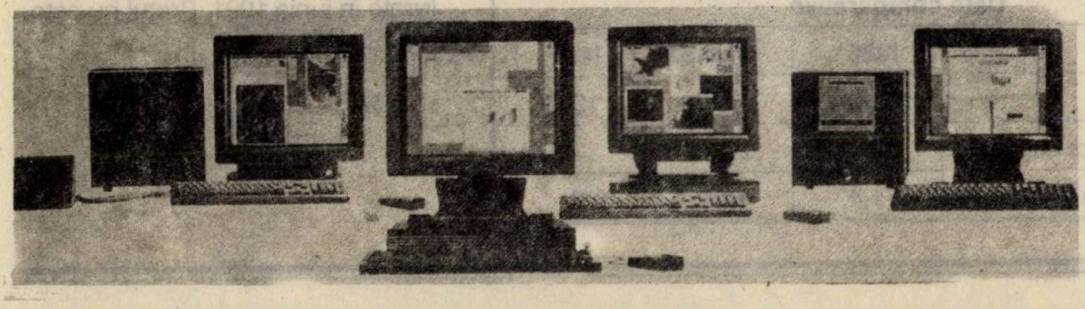
Fiecare Next dispune de un lector de dischete de 3,5" care lucrează cu "Extended Density" pe care se pot stoca de două ori mai multe date decît pe dischetele "High Density" ale calculatoarelor compatibile IBM; dischetele au o capacitate de 2,44 MByte şi costă 8 DM/bucată. Lectorul recunoaşte şi dischetele formatate şi înregistrate pe un calculator compatibil IBM, pe care le poate atît citi cît şi scrie fără să necesite în plus nici un program special (aşa cum este cazul calculatoarelor Macintosh).

Versiunile de bază "Next Station" şi "Next Station Color" sînt livrate cu harddisk-uri avînd capacitatea de 105 MByte (timp de acces 17 ms, rată de transfer 4 MByte/sec). Opţional poate fi livrat şi un harddisk de 340 Mbyte (timp de acces 15 ms, rată de transfer 8 MByte/sec). "Nextcube" poate fi echipat chiar şi cu un harddisk de 1,44 GByte (timp de acces 13 ms).

Puterea de calcul şi memoria ridicată recomandă utilizarea lui "Next Station" în special pentru aplicaţii grafice, CAD şi DTP: pe monitorul "Megapixel" (diagonală 17"=43 cm) poate fi vizualizată o pagină de format DIN A4 întreagă - doar monocrom - la o rezoluţie de 832x1120 pixeli, fiecare punct putînd fi reprezentat în 4 nuanţe de gri. "Next Station Color" însă la aceeaşi rezoluţie poate reprezenta 4096 culori.

Dacă totuşi nu este suficient, atunci cu "Nextcube" se pot obţine imagini şi mai împestriţate: pentru aceasta este necesară însă placa de extensie "Next Dimension" (cca. 8.700 DM) care permite reprezentarea a 16,7 milioane de culori (magistrala este de 32 biţi, 24 de biţi permit generarea a 16,7 milioane de culori, iar cei 8 biţi rămaşi permit reprezentarea a 256 de nivele de transparenţă), rezoluţia rămînînd aceeaşi.

Dar Next nu încîntă doar văzul ci şi auzul cu cipul Motorola special "56001 DSP" (DSP = Digital Signal Processor), utilizatorul poate înregistra, prelucra şi reda vorbirea, muzică şi zgomote; microfoanele şi difuzoarele stereo sînt înglobate în monitor. În acest fel utilizatorul



Familia Next.

poate înzestra documentele şi fişierele şi cu comentarii verbale.

Sisemul de operare utilizat de Next este Unix, sistem de operare pe care îl folosesc de altfel şi majoritatea celorlalte tipuri de staţii de lucru. Unix este recomandabil în special pentru lucrul în reţea şi pentru a permite rularea simultană a mai multor programe pe acelaşi calculator (multitasking). Dezavantajul: Unix este destul de incomod în operare. Pentru a contracara acest minus Next utilizează suprafaţa utilizator grafică "Nextstep" asemănătoare suprafeţei utilizator "Finder" a calculatoarelor Macintosh. Utilizatorul nu mai trebuie să tasteze comenzi complicate (cum trebuie s-o facă de ex. în cazul sistemului de operare MS-DOS) ci poate opera manipulînd simboluri cu ajutorul mouse-lui.

Next produce şi o imprimantă laser (8 pag./min.), care permite o rezoluţie selectabilă de 300 sau 400 dpi (dot per inch). Cu toate că imprimanta laser nu este o imprimantă Postscript (aşa se explică şi preţul de doar 3.900 DM), datorită faptului că sistemele dispun de soft-ul "Display Postscript" incorporat, imaginea de pe monitor va fi reprodusă în mod identic şi pe hîrtie (WYSIWYG = what you see is what you get).

Dar nici cel mai bun calculator şi nici cea mai bună suprafaţă utilizator nu-i sînt de nici un folos utilizatorului dacă nu există programare care să poată fi utilizate. Aceasta a fost una din cele mai serioase critici aduse primului calculator Next (vechiul "Nexcube"). Oamenii de la Next au încercat să elimine acest reproş, pe harddisk-ul fiecărui Next livrîndu-se în mod standard un număr de pachete soft.

În magazine există deja pachete soft produse şi de alte firme. În toate domeniile există deja programe interesante: procesoare de texte, grafică, Desktop Publishing, animaţie, matematică şi statistică, calcul tabelar, baze de date, programare.

Angajamentul unor firme de renume oferă garanţia faptului că nici în viitor nu va lipsi softul pentru Next: Wordperfect, Lotus, Ahton Tate şi Oracle se ocupă de dezvoltarea de programe pentru Next.

Model: Next-Station
Producător: Next Computers
Preţ: începînd de la 8200DM, (monitor monocrom 2100 DM, set accesorii 500 DM)
Procesor: Motorola 68040 (25 MHz)
Sistem de operare: Unix, Nextstep 2.0
Monitor: "Megapixel", rezoluţie 1120x832, 4 nivele de gri
Memoria principală: 8 MByte (max. 32 MByte)
Memoria de masă: 105 MByte (pînă la 400 MByte), lector de dischete de 3,5" (2.88 MByte pe dischete Extended Density)
Dimensiuni: 6,4x36,5x39,8 cm fără monitor. Greutatea 5,5 kg.

Cocluzionînd "Next Station" este un calculator foarte puternic, iar preţul de 10.800 DM (complet cu monitor) este pe măsură.

(R.M.)

Software pentru Next

Next livrează fiecare calculator cu o selecţie de programe cu care utilizatorul se poate apuca imediat de lucru. Cu "Next Stations" cu harddisk de 100 MB se livrează printre altele:

- **Nextmail**, cu ajutorul căruia utilizatorul poate expedia documente în reţea. Cu ajutorul unui modem-telefax acesta îşi poate utiliza Next-ul şi ca telefax.
- **WriteNow**, un procesor de texte (fără funcţie de scriere de scrisori în serie).
- **Draw**, un program de desenare.
- **Digital Webster**, un dicţionar englez, şi Webster's Thesaurus.
- **Date**, o agendă de lucru.

Cu toate calculatoarele avînd harddisk-uri mai mari, Next livrează şi alte soft-uri, printre altele şi un compilator **C**.

Vă prezentăm în continuare o selecţie de soft-uri pentru Next scrise de alte case de soft:

- **Wordperfect**, cunoscutul procesor de texte (Preţ: 500$).
- **Topdraw**, un superprogram de desenare al firmei Media Logic (Preţ: 595$).
- **Lotus Improv 1.0**, program de calcul tabelar, urmaşul lui Lotus 1-2-3, are toate şansele să impună un nou standard pentru programele din categoria sa (Preţ: 695$).
- **Soft PC**, emulator soft. Cu ajutorul lui pot fi rulate pe Next toate programele scrise pentru MS-DOS. (Preţ încă necunoscut).
- **Oracle**, sistem de gestiune a bazelor de date (Preţ încă necunoscut).
- **Framemaker**, un program Desktop-Publishing pentru Next (Preţ: 995$).

A calcula şi a prezenta

Quattro Pro 3.0

Borland încearcă să bată toate recordurile la prezentarea unor noi versiuni de programe. De-abia a fost lansat Quattro Pro 2.0 pe piaţă şi deja a fost prezentată noua versiune Quattro Pro 3.0.

A devenit deja un obicei ca la puţine săptămîni de la lansarea unui produs, să se prezinte o nouă variantă, în care să fie corectate "scamele" apărute şi care uneori era ornată cu cîteva noi extensii. Cu toate acestea ritmul, în care lucrează proiectanţii lui Quattro Pro, îţi taie respiraţia: în decurs de un an şi jumătate au fost produse două versiuni ale programului de calcul tabelar Quattro Pro. Ca şi în cazul versiunii anterioare, nici în cazul versiunii 3.0 nu este vorba doar despre modificări cosmetice ci de un program cu funcţiuni noi, impresionante, care-i v-a determina desigur pe utilizatori să opteze pentru noua versiune.

Pe lîngă suprafaţa orientată caracter este disponibil acum şi un mod de lucru WYSIWYG care permite virtualizarea tuturor fonturilor, a atributelor de scriere, a liniilor şi a umbririlor, aşa cum vor arăta după imprimarea lor pe hîrtie. Dar aceasta nu este totul, ecranul cu care produsul se prezintă în modul grafic a fost prelucrat în întregime. Etichetele inscripţionărilor liniilor şi coloanelor, butoanele meniurile şi cutiile de dialog sînt prezentate acum în trei dimensiuni.

Asemănător ca la Lotus 1-2-3, în modul grafic se poate face o scalare (zoom) cu un număr nelimitat de trepte între 25% şi 200%.

Deoarece vă prezentăm una din primele versiuni beta ale produsului, nu vă putem da încă date exacte referitoare la viteza de calcul. Ceea ce vă putem spune deja este că în modul WYSIWG Quattro Pro atinge o viteză acceptabilă. Defilarea pe ecran şi funcţiile la nivel de bloc au devenit mai lente, din cauza modului grafic, dar aceste dezavantaje se păstrează în limite acceptabile.

Şi la imprimarea unei tabele pe hîrtie sînt oferite acum posibilităţi de a scăpa de multe din necazurile obişnuite. Quattro Pro 3.0 oferă mai multe posibilităţi de a imprima pe hîrtie tabele complexe: dacă de exemplu tabela conţine mai multe linii şi coloane decît ar încăpea pe hîrtie, se poate apela funcţia "Print to fit" şi programul va imprima tabela la o astfel de scară încît aceasta să nu depăşească marginile hîrtiei. La cerere, raportul de micşorare poate fi stabilit şi manual. O tabelă poate fi şi mărită de pînă la 10 ori pentru a fi imprimată.

Această metodă ajută puţin, totuşi, atunci cînd este vorba despre tabele cu peste 100 de coloane. În acest caz poate fi folosită o nouă tehnologie integrată în Quattro Pro: asemănător ca în cazul auxiliarului (add-in) Lotus "Sideways", o tabelă poate fi listată ca "Banner". În acest caz ea este imprimată orizontal, fără întreruperi de pagină, astfel încît pe hîrtie fără sfîrşit (perforată) se poate obţine o copie 1:1 a tabelei.

Şi în editorul grafic au fost incluse noutăţi. Ca şi în cazul programelor de grafică profesionale desenul poate fi construit avînd la bază un rastru, care poate fi folosit pentru poziţionarea şi pentru dimensionarea obiectelor. Simultan poate fi activată şi funcţia "snap-to" caz în care Quattro Pro va începe unui obiect grafic întotdeauna exact într-un punct al rastrului şi îl va sfîrşi într-un alt punct al acestuia.

Ca şi pînă acum cîmpurile de text pot fi definite ca butoane şi la selectarea lor pot apela o altă diagramă sau un macrou. Nou este aşa- numitul buton "background" care poate fi de asemenea cuplat cu un alt grafic. Astfel este suficient un clic pe mouse într-un anumit loc de pe o diagramă pentru a fi afişată următoarea imagine.

Modul de prezentare al unei expuneri (slide-show) poate fi îmbunătăţit prin folosirea efectelor speciale din versiunea 3.0. La schimbarea unei imagini pot fi folosite efecte ca "defilare" sau "spargere" a imaginii pentru a capta atenţia spectatorului. Extrem de utilă este şi funcţia "autosave" în cazul graficelor: cu ajutorul ei putem fi siguri că orice modificare efectuată într-o diagramă va fi salvată pe harddisk.

Pe lîngă noutăţile enumerate, Quattro Pro 3.0 dispune şi de cîteva mici funcţii care în versiunea beta nu erau încă definitivate. În versiunea finală va fi posibilă atît citirea cît şi scrierea de fişiere CGM. În acest caz graficele vor putea fi preluate în toate programele de prezentare uzuale.

Meniul "File" a fost extins cu comanda "Save All", cu ajutorul căreia pot fi închise, cu o singură operaţie, toate fişierele deschise la un moment dat. Cîteva comenzi noi pentru construirea de macrouri permit însoţirea prezentărilor cu mici bucăţi muzicale.

Quattro Pro dovedeşte cu versiunea 3.0 că este un concurent serios în competiţia pe care o desfăşoară împreună cu Lotus 1-2-3 şi Excel, pentru titlul de cel mai bun program de calcul tabelar. Cerinţele hard (AT, 512 KByte RAM, harddisk) nu vor creşte nici în noua versiune.

(R.M.)

Tabel comparativ

	Quattro Pro 3.0	Excel 3.0	Wingz
Preţ	495 DM	1847 DM	1704 DM
Date tehnice			
Nr. linii X Nr. coloane X Nr. foi de lucru	8192 x 256 x 0	16384 x 256 x 0	9999 x 256 x 0
Fişiere deschise simultan	32	funcţie de mem.	funcţie de mem.
Foi de calcul tridimensionale	nu	nu	nu
Lungime cîmp	255 caractere	255 caractere	255 caractere
Coprocesor	da	da	da
Spaţiu ocupat pe harddisk	4,5 MByte	5 MByte	3 MByte
Caracteristici de putere: Calcul			
Funcţii autodefinite	nu	da	da
Căutare pt. obţinerea unei valori finale impuse	da	da	nu
Solver	da	da	nu
Tabele ce-se-întîmplă-dacă cu X variabile	2	2	2
Consolidare inteligentă	nu	da	nu
Cuplare cu fişiere de pe harddisk	da	da	nu
Caracteristici de putere: Publicistică			
Combinare de tabele şi diagrame	da	da	da
Suport Postscript	da	da	da
Import formate grafice	CLP, CGM	clippboard	clippboard
Funcţii de desenare în tabelă	nu	da	da
WYSIWYG	da	da	da
Previzualizare	da	da	da
Imprimare la dimensiunea hîrtiei	da	da	nu
Caracteristici de putere: Baza de date			
Mască pentru baza de date	nu	da	nu
Chei de sortare	5	3	255
Interogare bază de date externă	da	da (cu Q+E)	nu
Actualizare bază de date externă	nu	da (cu Q+E)	nu
Compatibilitate cu Lotus 1-2-3			
Structura meniurilor	da	Help special	nu
Taste	da	nu	nu
Macrouri	da	cu conversie	nu
Import / Export			
WKS	da	da	da
WK1	da	da	da
WK3	nu	da	nu
DBF	da	da	da
SYLK	da	da	da
DIF	da	da	da
BIFF	nu	da	da

Computer Aided Design

Zilele planşetei de desen sînt numărate. Pentru efectuarea de desene tehnice PC-urile cu soft corespunzător au nevoie de mai puţin loc, risipesc mai puţină hîrtie şi oferă utilizatorului mai multă putere şi confort decît creionul.

Prescurtarea CAD este folosită pentru Computer Aided Drafting, şi respectiv pentru Computer Aided Design. A doua variantă a acestei noţiuni este mai cuprinzătoare şi mai uzuală. În timp ce Drafting se referă la desenarea cu ajutorul calculatorului, Design depăşeşte cu mult cadrul lucrărilor grafice efectuate cu ajutorul calculatorului. Din acest motiv ar fi greşit să traducem această noţiune doar prin construcţie: ea cuprinde mult mai mult şi anume întregul spectru al activităţilor care trebuiesc depuse pentru crearea şi întreţinerea unui produs.

Sistemele CAD de azi au, de exemplu, ca sarcini tipice pe lîngă desenare şi construire, şi simularea unor încărcări, mişcări şi caracteristici de material. Programele trebuie să pună la dispoziţie şi unelte pentru tratarea desenelor şi a listelor componentelor din structură.

Un element esenţial care trebuie să ghideze decizia achiziţiei unui sistem CAD este şi compatibilitatea formatului datelor cu a altor aplicaţii. De aceasta depinde posibilitatea ca utilizatorii să poată prelua sau preda desenele lor din/spre alte sisteme de calcul.

De cînd calculatoarele personale au devenit atît de puternice şi de accesibile ca preţ încît pe ele se pot rezolva sarcini complexe cu un efort acceptabil, CAD s-a impus în multe domenii. Cele mai importante domenii de utilizare sînt: construcţii de maşini, arhitectură, electrotehnică şi mai nou din ce în ce mai mult arhitecura de interior. Producerea de softuri pentru crearea de construcţii 3D şi umbriri şi posibilităţile care decurg din acestea, de a crea şi calcula imagini spaţiale expresive de produse, clădiri sau peisaje, au propulsat CAD-ul în vîrful aplicaţiilor profesionale de grafică.

Diferenţa dintre un program CAD şi un program de desenare constă în modul în care sînt tratate componentele individuale ale desenului. Programele CAD lucrează orientat obiect. Calculatorul memorează deci pentru fiecare obiect desenat descrierea sa matematică şi dimensională exactă şi îi ataşează diferite atribute, cum ar fi de pildă starea, culoarea, tipul, nivelul (layer), care îi stabilesc apartenenţa la o anumită grupă de obiecte. În plus mai pot fi memorate date despre materiale, firmele furnizoare, indicaţii de prelucrare, etc.

Un desen CAD este deci o bază de date matematică din care poate fi apelat oricînd un obiect pentru a i se face adăugiri (inserări), modificări, analize şi multe altele.

Programele de desenare nu răspund tuturor acestor necesităţi. În cazul lor imaginea este memorată într-un fişier ca o imagine rastru, punct cu punct. Elementele individuale nu pot fi identificate.

Fiecare sistem CAD se compune din cinci componente de bază fără de care un astfel de program nu ar putea lucra:

- masca ecran
- unitate de intrare (introducere date)
- reprezentarea desenului
- baza de date CAD
- ieşire şi export

Masca ecran

Monitorul (ecranul) este împărţit în mai multe domenii de lucru, care trebuie să rezolve probleme diferite. Domeniul principal este întotdeauna domeniul grafic: în el se desenează. De jur împrejurul lui se ordonează cîmpurile de comandă, care sînt rînduite ca liste text la marginea superioară sau inferioară a ecranului, ca şi la marginile sale laterale.

Deoarece un program CAD are foarte multe comenzi, cel mai adesea meniurile sînt aranjate pe mai multe nivele şi trebuie "răsfoite" pentru a putea apela comanda dorită.

Diferitele programe utilizează tehnici de operare diferite. O anumită normare a modului de operare transpare doar de curînd, din momentul în care anumite programe au fost transpuse în mediul Windows 3.0. Întrucît modul de operare sub această suprafaţă utilizar este precis stabilit, toate programele care funcţionează sub acest mediu vor avea o deservire asemănătoare.

Din punct de vedere hardware pretenţiile referitoare la monitor şi la adaptorul grafic sînt ridicate. Deoarece în urma ocupării unei părţi de ecran cu cîmpurile de control rămîne relativ puţin spaţiu pentru desenul prpriu-zis, diagonala ecran trebuie să fie cît mai mare. Dimensiunea ei este limitată însă din considerente financiare şi tehnice. Standard sînt utilizate monitoare de 19" cu o rezoluţie de 1024x768 puncte. În acest caz se poate lucra profesional. Costurile se păstrează şi ele între limite acceptabile. Dacă însă se desenează numai din cînd în cînd, atunci 10.000 DM este mult prea scump. În acest caz poate fi folosit

mulţumitor şi un monitor de 14" cu un adaptor grafic VGA.

Unitatea de intrare

În timp ce pe ecran sînt prezentate diferite domenii de control, pentru introducerea datelor posibilităţile sînt limitate. Cea mai simplă, dar şi cea mai incomodă, este introducerea datelor de la tastatură. Cu toate acestea pentru introducerea valorilor numerice nu ne putem dispensa de tastatură. În plus aproape toate programele CAD folosesc tastatura ca ultimă soluţie în cazul unor erori sistem.

Un mod de comandă mai comod este controlul cursorului prin intermediul mouse-ului. Aproape toate programele CAD permit operarea cu un mouse. Marele avantaj al acestui mod de operare constă în faptul că în acest caz privirea operatorului nu trebuie să se plimbe între mai multe aparate.

Tocmai această comutare a privirii de pe ecran pe aparatul de introducere este singurul dezavantaj al unui aparat de introducere utilizat în domeniul profesional: tabela digitizoare cu lupă. Pe aceasta pot fi aplicate măşti specifice soft, pe care să fie reprezentate simbolic toate comenzile.

Reprezentarea desenului

Desenarea este sarcina principală a unui sistem CAD. Utilizatorul trebuie să aibă la dispoziţie comenzi de desenare elementare care să permită desenarea de linii, curbe, cercuri, poligoane, corpuri 3D elementare sau texte. În plus mai trebuie să existe comenzi de uşurare a lucrului; cum ar fi: "capturarea" cursorului într-un anumit punct, la intersecţia a două linii. Doar în acest mod pot fi executate desene exacte şi la scară.

Mai sînt şi alte comenzi care pot uşura lucrul: elementele desenate trebuie să poată fi deplasate, copiate, oglindite, întinse sau scurtate. Dar aceasta nu este totul: caracteristicile unui element pot fi modificate, mai multe elemente se pot grupa fiind tratate în continuare ca un singur element, elementele grupate pot fi separate din nou, sau pot fi utilizate în continuare ca macrouri, şi, şi...

Această înşiruire ar putea continua la nesfîrşit, dar se poate observa deja cît de complex este un soft pentru CAD şi cîte exerciţii sînt necesare pînă să se înveţe modul de operare.

Baza de date CAD

Toate elementele şi respectiv obiectele unui desen sînt memorate şi prelucrate de către programul CAD într-un anumit format. Printre aceste prelucrări se numără, în primul rînd, regăsirea unei construcţii deja desenate. Pentru ca acest lucru să fie posibil trebuie respectate anumite convenţii la creare şi memorare (stocare): la stocarea unui desen pe lîngă numele acestuia mai trebuie stabilite şi alte elemente de bază, cum ar fi formatul hîrtiei, unităţile de măsură, nivelul, tipul liniilor, directorul, ş.a.m.d.

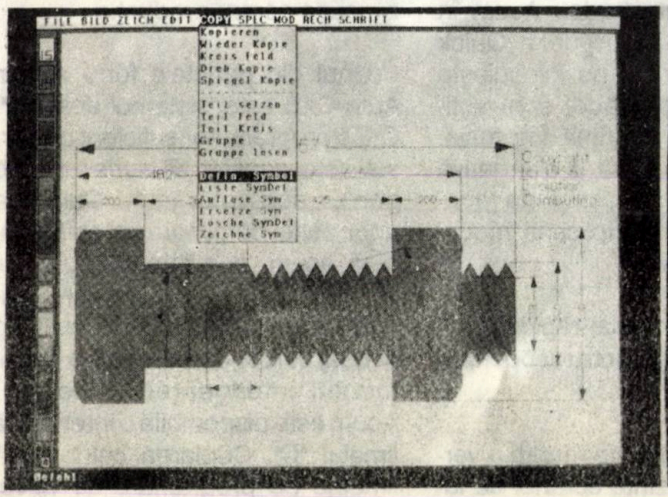
Ieşire şi export

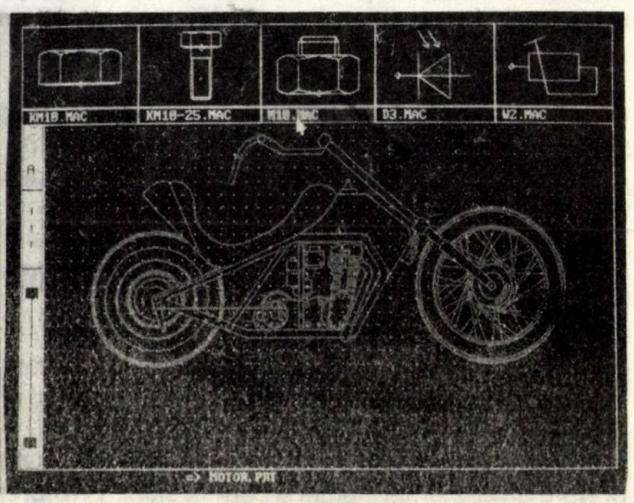
Rezultatul unui proces de desenare sau de construcţie, în zilele noastre, mai trebuie încă să fie prezentat pe hîrtie. Pentru reprezentarea pe hîrtie sînt disponibile diferite variante: pentru utilizări profesionale şi industriale se utilizează un plotter, pentru amatori este suficientă o imprimantă.

A doua interfaţă importantă spre lumea exterioară o reprezintă formatele de ieşire (export) cunoscute de program. Există mai multe formate de ieşire standard, recunoscute de majoritatea programelor: DXF (format specific Auto CAD), IGES şi HPGL.

Există o paletă foarte bogată de programe CAD, cu performanţe, dar şi la preţuri, mult diferite. O scurtă enumerare: Easycad 2 (565 DM), Megacad 2.0 (598 DM), Designcad (1140 DM), Generic CADD Level 3 (2000 DM), Geddy Cad (515 DM), Caddy Junior (789 DM), Powercad 1.23 (1298 DM), Autosketch 2.0 (350 DM), sub Windows: Drafix Win-CAD 1.1 (2274 DM), Technobox CAD/2 1.2 (2498 DM). Numărul 1 în top, cel care impune standardul, este Auto CAD program a cărui versiune 11.0 v-o prezentăm în continuare.

(R.M.)

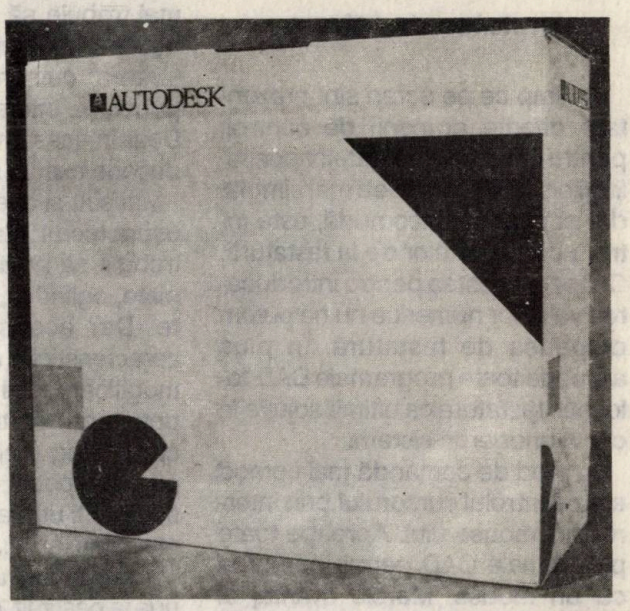




Lucrurile bune se obţin cu răbdare

Test: Auto - CAD 11.0

Versiunea 11 a clasicului "Auto-CAD" a fost anunţată încă în urmă cu un an. Acum este disponibilă, în sfîrşit, prima versiune, fapt care ne prilejuieşte prezentarea unui prim test al produsului.



Auto-CAD 11.0: Nu numai designul ambalajului este nou, şi în program s-au făcut multe modificări

Producătorii de soft se joacă cu plăcere "de-a şoarecele şi pisica": nici nu apucă bine cineva să prezinte un produs nou sau îmbunătăţit, că deja concurenţa anunţă un program încă şi mai bun. Penibil este însă că atunci cînd cu toată vîlva care se face, sertarul este gol. Ceva asemănător s-ar fi putut întîmpla şi în cazul producătorului de soft Autodesk. Acest lucru ar explica, în orice caz, de ce între anunţarea şi momentul lansării pe piaţă a versiunii "Auto-CAD 11.0" a trecut aproape un an.

Ca întotdeauna, se pare că aşteptarea a meritat. Abundenţa noutăţilor versiunii Auto CAD 11.0 este de luat în seamă. Pe primul loc ar fi de semnalat facilităţile 3D. Cuvîntul magic pentru acestea este "AME" (Advanced Modeling Extension). În spatele acestei noţiuni se ascunde o nouă procedură de tratare a construcţiilor tridimensionale orientate pe volume, un progres semnificativ faţă de modelele anterioare orientate pe muchii şi suprafeţe.

Diferenţa între reprezentările 3D ale versiunilor 10 şi 11 probabil că vor deveni mai clare în urma unui exemplu: pînă acum un corp putea fi creat doar asemănător modului în care se face o construcţie din hîrtie. Din afară arăta ca un corp "adevărat" dar în interior nu se putea privi. Noua metodă - orientată pe volume - descrie corpurile şi în interior. Întregul mediu cu geometria sa, caracteristicile de material şi rezultînd de aici modul de comportare al rezistenţei inerţiei şi centrului de greutate, sînt descrise prin metoda AME.

Acum, cu ajutorul unei comenzi, pot fi afişate toate valorile relevante şi pot fi preluate pentru calcul în aplicaţiile învecinate. În afară de acestea se pot executa secţiuni la înălţimea dorită. Construcţiile complicate pot fi realizate prin îmbinarea unor corpuri simple şi la nevoie pot fi descompuse din nou.

Strîns legată de reprezentarea 3D este şi posibilitatea de umbrire a suprafeţelor. Pînă acum Auto-CAD dădea posibilitatea pregătirii unor astfel de "umbriri", dar pentru umbrirea propriu-zisă era necesar programul "Autoshade". Acum în Auto-CAD există funcţiunea "Quick Shade", care deşi nu înlocuieşte programul Autoshade, este suficientă pentru o primă impresie. "Quick Shade" oferă o alternativă binevenită, pentru comanda "acoperit", tocmai la complicatul "model plasă din sîrmă".

Umbrirea este mai rapidă decît calculul necesar pentru acoperirea (îmbrăcarea) liniilor.

O altă noutate importantă a versiunii 11 o prezintă ecranul de lucru. Acum există posibilitatea definirii unor aşa-numite domenii model şi hîrtie. În domeniul model este creat desenul prototipului. Aici este prezentată descrierea completă a modelului volumetric, indiferent că este vorba despre un model mic cît o cutie de chibrituri, sau mare cît un teren de fotbal. Cu totul altfel se prezintă situaţia în domeniul hîrtiei. Aici pot fi definite sau mascate diferite ferestre care să conţină numai anumite detalii ale modelului.

Ferestrele pot fi aduse împreună pe formatul de hîrtie dorit şi listate prin intermediul unui plotter sau a unei imprimante. În acest mod pot fi create cu uşurinţă desene de execuţie pentru producţie.

Unul din punctele forte ale lui Auto-CAD este "sistemul deschis". Orice programator suficient de versat este capabil să scrie utilitare pentru acest program. Cuplarea unor astfel de programe se făcea, pînă acum, prin intermediul unui limbaj propriu de programare "Autolisp", un limbaj înrudit cu Lisp (limbaj utilizat cu precădere în domeniul inteligenţei artificiale). Acum este disponibilă o interfaţă la limajul "C". Cuplarea celor două limbaje de programare se dove-

deşte a fi aproape ideală. Acest lucru fiind mijlocit de "ADS" (Auto-CAD Development System). Prin intermediul ADS pot fi integrate programe externe, indiferent de faptul că ele sînt programe proprii, scrise de utilizator, sau sînt pachete soft deja existente, cum ar fi: programe de calcul tabelar, baze de date sau soluţii specifice branşei.

Auto-CAD 11.0 poate fi utilizat şi în reţea, funcţionînd pe toate sistemele de operare de reţea, cum ar fi Novell, 3Com sau NFS. Posibilitatea de a fi instalat şi pentru un post individual de lucru, a rămas. În cazul ambelor moduri de instalare este disponibilă o procedură de logare (de deschidere a unei sesiuni de lucru) extinsă. Acum pot fi definite parole care să dea certitudinea că la anumite date au acces numai utilizatorii specificaţi. De asemenea pot fi blocate fişierele, ceea ce dă siguranţă că un anumit desen nu este prelucrat simultan de mai mulţi utilizatori. Extensiile pentru lucrul în reţea implică un efort de instalare mai mare, dar o parte semnificativă a acestui efort este preluată de procedura de instalare.

Ţinînd cont de multitudinea noutăţilor, nu este de mirare că Auto- CAD 11.0 nu poate fi rulat decît pe calculatoare cu 386 cu coprocesor şi respectiv pe calculatoare cu 486. Memoria minimă necesară este de 2 MByte. Dacă se doreşte utilizarea AME sau a funcţiunilor de umbrire atunci sînt necesari cel puţin 4 MByte. Auto-CAD 11.0 ştie să lucreze şi cu memorie virtuală pe harddisk, dar aceasta este mai lentă decît memoria RAM. Deci: cu cît există mai multă memorie, cu atît este mai bine. Auto-CAD recunoaşte standardul VCPI (Virtual Control Program Interface). Acest standard include convenţiile stabilite de mai mulţi producători pentru stabilirea modului în care sistemul de operare DOS trebuie să trateze memoria "extinsă" şi "expandată". Pentru aceasta sînt necesare drivere de memorie care să respecte standardul VCPI. Dintre acestea enumerăm: "Compaq CEMM 4.01", "Quarterdeck QEMM 4.01" şi "386 - MAX" al firmei Qualitas System.

Concluzionînd aşteptarea lui Auto - CAD 11.0 a meritat. Noile facilităţi nu sînt dedicatee în primul rînd utilizatorilor obişnuiţi, dar le sînt indispensabile utilizatorilor profesionali.

Preţ: Auto-CAD 11.0 7296 DM, cu AME 7866 DM.

(R.M.)

Lexicon

Element Cea mai mică unitate de desen, de ex. o linie, un arc de cerc sau un cerc

Fişier DXF Format fişier pentru schimbul de desene între diferite programe CAD

Grid Rastru. Pe monitor este afişat un rastru pentru orientare. Acesta nu va fi listat pe plotter sau imprimantă

Grupă Obiectele individuale pot fi incluse în grupe pentru a putea fi prelucrate împreună. Grupările pot fi descompuse din nou, la nevoie.

Hatch Haşură. Haşurarea trebuie să respecte normele DIN

HPGL Limbaj de comană pentru controlul plotter-elor (Hewlett Packard Graphics Language).

Layer Nivel desen: asemănător modului de lucru cu folii pe un diaproiector. Un nivel poate fi mascat, aceasta înseamnă că nu va fi afişat pe ecran.

Listă de componente Lista simbolurilor care apar într-un desen şi cantitatea în care apar.

Ortho Nu mai pot fi desenate decît linii orizontale sau verticale. Unele programe permit specificarea unui unghi cu care să se lucreze în continuare.

Pan (Panorama) Deplasarea secţiunii vizibile din desen

Plotter Masă de desen automată. Aparat de ieşire cu ajutorul căruia desenul este reprezentat în tuş, sau carioca, pe hîrtie

Polyline Trasarea unei linii care constă din mai multe segmente de linii, cu o singură operaţie

Printplot Reprezentarea desenului pe o imprimantă de rezoluţie ridicată. Nu se atinge însă rezoluţia unui plotter.

Simbol (Macrou) Detaliile utilizate frecvent pot fi memorate ca macrouri şi ulterior pot fi inserate la nevoie şi în alte desene.

Snap Cursorul poate fi poziţionat numai în anumite puncte definite prin valoarea de prindere (snap). În acest mod se poate face poziţionarea exactă pe un nod (o intersecţie), fără a fi necesară specificarea manuală a coordonatelor punctului respectiv.

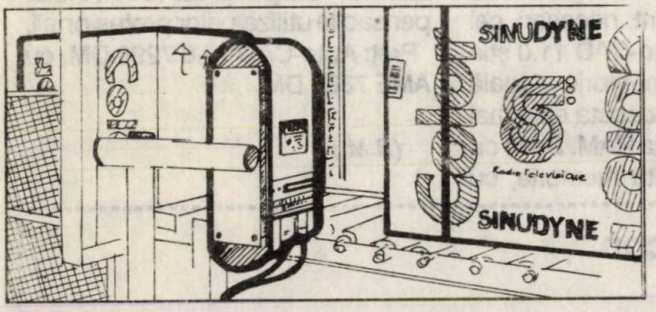
Varia Scara unui obiect poate fi corelată cu scara restului desenului. Această operaţie este necesară atunci cînd într-un desen se inserează macrouri (obiecte) predefinite.

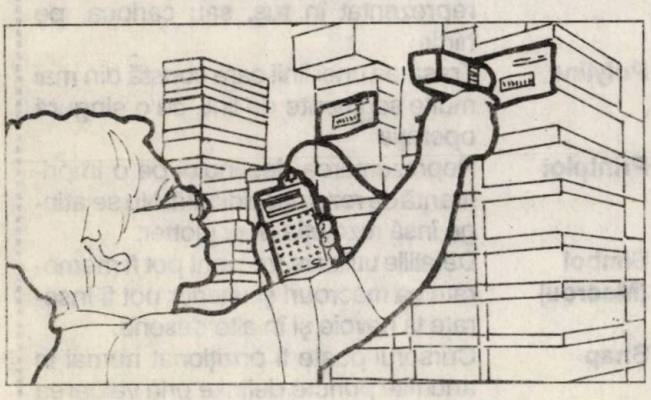
Zoom Metodă folosită pentru a permite prezentarea unui desen în detaliu. Factorul de scalare este selectabil. Dacă este mai mare decît 1 desenul va fi mărit şi vor fi afişate mai multe detalii, dacă este mai mic decît 1 desenul va fi micşorat şi se va obţine o vedere de ansamblu mai bună.

Cod de bare

Printre cele mai mari probleme ale întreprinderilor industriale şi comerciale moderne se numără şi felul în care funcţionează magaziile, felul în care se realizează fluxul tehnologic. Cît anume din timpul de realizare al unei maşini mai complexe este timp de lucru/asamblare efectiv şi cît anume este timp "de aşteptare" în diverse faze ale trecerii respectivului produs prin fabrică este un factor pînă la urmă extrem de important, care se reflectă direct atît în productivitate cît şi în preţ. Cerinţele din ce în ce mai dure în această direcţie au determinat dezvoltări în două direcţii principale:

- utilizarea unor sisteme de transport moderne





- utilizarea calculatoarelor la urmărirea fluxului de materiale

Pentru o utilizare consecventă a calculatoarelor în urmărirea fluxului de materiale în toate fazele, este inevitabilă identificarea directă a încărcăturii transportate, în fiecare ipostază (pachet, palet, container etc.).

Datorită progreselor însemnate din opto-electronică, în prezent este posibilă citirea automată a datelor înregistrate (direct) pe încărcătura transportată şi transmisia lor, printr-o interfaţă standard, spre calculator, pentru a fi prelucrate. Datele de identificare de pe fiecare încărcătură trebuie să fie codate sau înregistrate într-un fel oarecare; cel mai utilizat sistem, în prezent este codul de bare. Datele înregistrate sub formă de cod de bare se tipăresc de regulă direct pe ambalaje, avize de expediţie, fişe de magazie, bilete de avion, legitimaţii, etichete etc.

În repaus, datele astfel codate pot fi citite manual, folosind un "creion de citire" pentru a le prelua şi disponibiliza calculatorului. Unde acest lucru nu are sens - de exemplu la instalaţii unde se lucrează la bandă, cu încărcături mobile - datele de identificare pot fi preluate (de la o anumită distanţă) prin intermediul unor laser - scannere cu He şi Ne sau al unor camere de luat vederi CCD.

Preluarea automată a datelor tipărite oferă utilizatorului diverse posibilităţi:

- generarea tuturor documentelor ce privesc depozitarea şi expediţia (aviz de expediţie, fişă de urmărire, chitanţă, etc.)
- controlul automat al întregului sistem de transport şi depozitare
- inventar permanent la zi şi (prin aceasta) date exacte relativ la stocurile existente (de produse şi/sau materii prime); ca o bază pentru deciziile conducerii.

Nenumărate sînt exemplele de utilizare ale aplicaţiilor construite pornind de la codul de bare:

- în biblioteci (şi/sau videoteci): toate cărţile sînt identificate prin intermediul unui cod de bare. De asemenea, fiecare utilizator are o legitimaţie, identificată printr-un cod de bare. Toate împrumuturile se fac prin intermediul calculatorului, fără hîrtie.
- în depozitele de componente electronice (sau alte articole mărunte) intrările şi ieşirile de marfă se înregistrează prin intermediul calculatorului. Prelucrarea codului fiecărui articol se face prin intermediul codului de bare, aplicat pe ambalaje, şi al "creionului de citire".
- legitimaţiile sau abonamentele se recunosc folosind cititoare cu fantă; astfel, se poate da acces la calculatoare, intrare la biblioteci, schi-lift, domenii militare etc.
- pe benzi de fabricaţie, codurile de bare aplicate pe cartoane permit preluarea datelor - prin laser-scanner montat fix, într-o anumită poziţie, astfel încît citirea să

se facă de la o distanţă de 200 - 800 mm - şi transmiterea lor la nivelul superior.

Codarea

La concurenţă cu alte sisteme de codare care să permită identificarea automată sau semiautomată a diverselor entităţi ce trebuiesc urmărite, codul de bare are un avantaj indiscutabil: este relativ simplu de produs şi relativ simplu de recunoscut.

Sistemul OCR se bazează pe recunoaşterea caracterelor alfabetice cu care sîntem obişnuiţi, cu restricţia că forma caracterelor trebuie să fie standardizată pentru a permite citirea automată. Avantajul claritătii - datele nu trebuiesc codate - este anulat de aparatele de citire şi recunoaştere complicate, precum şi de cerinţele ridicate vizavi de calitatea imprimării. La acesta se adaugă rate de erori mari: cercetări făcute în aplicaţiile tip supermarket au evidenţiat peste 30% citiri nereuşite din prima încercare şi peste 1% citiri eronate.

Sistemul magnetic cere şi el un efort tehnic şi financiar deosebit. Cînd codul trebuie produs în cantităţi numerice importante, limita economicităţii este foarte repede depăşită.

Codul de bare poate fi aplicat direct pe orice marfă (tub, carton, cutie, sticlă, etc.) sau poate fi aplicat ulterior, ca etichetă. În domeniul alimentar, codul EAN

Terminologie

Linie elementul întunecat al unui cod

Spaţiu elementul deschis dintre două linii ale unui cod

Interspaţiu spaţiul dintre ultima linie a unui caracter şi prima linie a următorului caracter în cadrul unui cod de baze discret

Element expresie folosită pentru a descrie o linie sau un spaţiu component al unui cod

Modul elementul cel mai îngust dintr-un cod. Liniile sau spaţiile mai late se calculează ca multiplii ai modulului

Lărgimea modulului X: defineşte lărgimea celui mai îngust element (de ex. X=0,33 mm) Zona "de linişte" - zona de culoare deschisă dinainte de caracterul de start şi de după caracterul stop. Zona de linişte este necesară pentru a da cititorului informaţia de "început de cod". Minimul este de 10 ori lărgimea modulului, dar cel puţin 2,5 mm. La aplicaţii ce folosesc scannere cu un domeniu de contrast mai mare, zona de linişte trebuie aleasă mai mare (L = 15 x lărgimea modulului, dar cel puţin 6,5 mm).

Simbol în cod de bare un cod de bare complet, începînd cu o zonă deschisă înaintea codului, un caracter de start, unul sau mai multe caractere utile, un caracter de stop şi încă o zonă deschisă. Dedesubt, un rînd cu textul în clar.

Caracter de start/stop - orice cod începe cu un caracter de start şi se termină cu un caracter de stop. Astfel citirea se poate face în ambele direcţii.

Cod cu autoverificare - cod de bare care permite verificarea fiecărui caracter, după un algoritm prestabilit. Erorile de substituţie sînt posibile

Cod discret cod de bare în care fiecare semn începe şi se termină cu o linie. Interspaţiul nu face parte din cod.

Cod continuu cod de bare la care interspaţiul face parte din cod

Cifră de control - unul sau mai multe semne suplimentare, care permit recunoaşterea erorilor

Eroare de substituţie - un semn e înlocuit de un altul. Numai prin utilizarea unei cifre de control, acest tip de eroare poate fi aproape exclus

Citire nereuşită - decodorul nu este în stare să decodifice informţia citită

Citire falsă informaţia decodată nu coincide cu cea din codul de bare (eroare de substituţie).

Rata de primă citire - numărul de citiri corecte la prima încercare, împărţit la numărul de încercări

Cod de fotoimprimare - prin fotoimprimare, pot fi produse coduri de orice densitate

Cod de densitate mare - cod foarte strîmt, produs prin fotoimprimare sau cu imprimante cu ciocănele

Cod de densitate medie - cod produs cu imprimante matriceale, termice sau cu jet de cerneală (modul 0,3 - 0,5 mm)

Cod de densitate mică - cod produs cu imprimante matriceale, termice sau cu jet de cerneală (modul 0,5 mm).

a devenit practic standard european; în domeniul medico-chimical, s-a afirmat codul CODABAR, iar codurile Code 2/5 5 linii sau codul 2/5 interleaved se utilizează în mai toate domeniile industriale.

Cele mai multe din codurile uzuale se bazează pe un principiu binar, cu un număr de linii/spaţii înguste sau largi. Secvenţa acestor linii respectiv spaţii determină un anumit conţinut numeric sau chiar alfanumeric.

Citirea se face optic. Prin reflexia diferenţiată a liniilor negre şi spaţiilor albe, în receptorul optic apare un tren de impulsuri care corespunde acestei secvenţe de linii şi spaţii.

Prin codare acest tren de impulsuri este apoi interpretat ca secvenţă de date.

Mărimea codului este determinat în principiu de parametrii aplicaţiei - de exemplu, felul caracterelor de reprezentat (numerice, alfanumerice) prin numărul de caractere, spaţiul disponibil, posibilităţile de imprimare, etc.

Toţi aceşti factori au făcut ca în decursul timpului pentru diverse aplicaţii să apară o multitudine de coduri de bare, care toate îşi au (sau respectiv aveau) justificarea. Cîteva din cele mai utilizate coduri, inclusiv tabelele de adevăr corespunzătoare, sînt prezentate în paginile următoare.

Cod 2/5 5 linii industrial

Generalităţi - cod numeric; pot fi reprezentate cifrele 0-9. Acest cod e construit cu două linii late şi trei linii înguste. Raportul de imprimare linie îngustă:linie lată (R) este de 1:2 pînă la 1:3. Spaţiile nu conţin informaţii.

Avantaje - codul constă numai din linii, în spaţii nu este cuprinsă informaţie. Toleranţa e mare (15%), de aceea poate fi produs prin cele mai simple procedee de imprimare

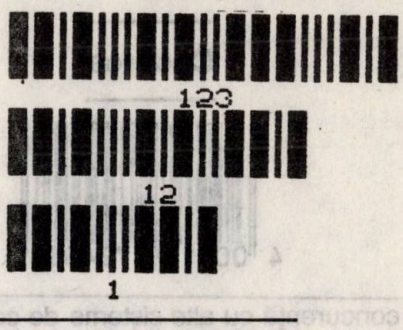
Dezavantaje - densitate mică a informaţiei (4,2 mm/cifră la o lăţime a modulului x = 0.3mm şi R = 1:3.)

Tabela de coduri:

Caracterul	L1	L2	L3	L4	L5
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
0	0	0	1	1	0
start	1	1	0		
stop	1	0	1		

L1-L5 = Linia 1 - 5
1 = Linie lată
0 = Linie îngustă

Exemplu:



Cod 2/5 interleaved

Generalităţi - cod numeric; pot fi reprezentate cifrele 0-9. Acest cod e construit cu 2 linii late şi 3 linii înguste. Raportul de imprimare (R): linie îngustă: linie lată 1:2 pînă la 1:3, spaţiu îngust: spaţiu lat 1:2 pînă la 1:3. Dacă cel mai îngust element e mai mic de 0,5 mm, atunci linie (spaţiu) îngustă" linie (spaţiu) lat = 1:2,25. Prima cifră este reprezentată prin 5 linii, iar a doua cifră prin spaţiile ce urmează nemijlocit liniilor primei cifre:

Avantaje: densitate mare de informaţie. (2,7 mm/cifră la o lărgime a modulului de x 0,3 mm şi un raport de imprimare R = 1:3); autoverificabil

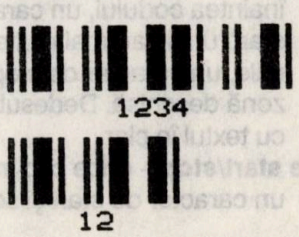
Dezavantaje: toate spaţiile poartă informaţie, de aceea toleranţa trebuie să fie mai mică de 10%.

Tabela de coduri:

Caracterul	L1	L2	L3	L4	L5
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
0	0	0	1	1	0
start	0	0			
stop	1	0			

L1-L5 = Linia / Spaţiul 1 - 5
1 = Linie/Spaţiu lat(ă)
0 = Linie/Spaţiu îngust(ă)

Exemplu:



Code 2/5 3 linii (Matrix)

Generalităţi - cod numeric; pot fi reprezentate cifrele 0-9. Fiecare caracter constă din 5 elemente, 3 linii şi două spaţii. Două elemente sînt late şi 3 sînt înguste. Deci atît liniile cît şi spaţiile pot fi late sau înguste. Interspaţiul nu poartă informaţie. Raportul de imprimare (R) linie îngustă: linie lată = 1:2,25 pînă la 1:3; spaţiu îngust: spaţiu lat = 1:2,25 pînă la 1:3. Linie deosebit de lată în caracterul de start şi cel de stop.

Avantaje: - densitate mare de informaţie (3,0 mm/cifră la o lăţime a modulului x = 0,3 mm şi R = 1:3) Nu există informaţie între cifre

Dezavantaje: toleranţă <= 10% Caracter supralat în informaţia de start/stop.

Tabela de coduri:

Caracterul	L1	S1	L2	S2	L3
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
0	0	0	1	1	0
start	11	0	0	0	0
stop	11	0	0	0	0

L1-L3 = Linia 1 - 3, S1-S2 = Spaţiul 1 - 2
1 = Linie/Spaţiu lat(ă)
0 = Linie/Spaţiu îngust(ă)
11 = 1,5 x Linie lată

Exemplu:

Cod 2/5 3 linii (Datalogic)

Generalităţi - cod numeric, pot fi reprezentate cifrele 0-9. Fiecare caracter constă din 5 elemente, 3 linii şi două spaţii. Două din elemente sînt late şi 3 sînt înguste. Deci atît liniile cît şi spaţiile pot fi late sau înguste. Interspaţiul nu poartă informaţie. Raportul de imprimare (R): linie îngustă: linie lată = 1:2,25 pînă la 1:3; spaţiu îngust: spaţiu lat = 1:2,25 pînă la 1:3

Avantaje - densitate mare de informaţie (3,0 mm/cifră la o lăţime a modulului x = 0,3 mm şi R = 1:3) Nu există informaţie între cifre

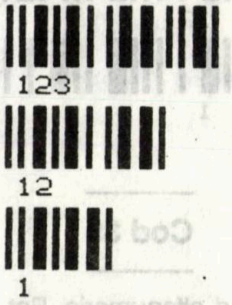
Dezavantaje - toleranţă mai mică de 10%.

Tabela de coduri:

Caracterul	L1	S1	L2	S2	L3
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
0	0	0	1	1	0
start	11	0	0	0	0
stop	11	0	0	0	0

L1-L3 = Linia 1 - 3, S1-S2 = Spaţiul 1 - 2
1 = Linie/Spaţiu lat(ă)
0 = Linie/Spaţiu îngust(ă)

Exemplu



Codabar

Generalităţi - cod numeric cu 6 caractere speciale. Pot fi reprezentate cifrele 0-9, minus (-), dolar ($), două puncte (:), slash (/), punct (.), plus (+). Fiecare caracter constă din 7 elemente, 4 linii şi 3 spaţii. Se folosesc 2 sau 3 elemente late şi 5 sau 4 elemente înguste pentru reprezentarea codului. Interspaţiul nu poartă informaţie. Raportul de imprimare: (R) linie îngustă: linie lată (1:2) pînă la 1:3; spaţiu îngust: spaţiu lat 1:2 pînă la 1:3 Dacă elementul îngust e mai mic de 0,5 mm, atunci element îngust: element lat = 1:2,25.

Avantaje: în afară de cifre, pot fi reprezentate încă 6 caractere speciale. Interspaţiul nu poartă informaţie.

Dezavantaje: densitate de informaţie redusă. (5,5 mm/cifră la o lăţime a modului de x = 0,3 mm şi R = 1:3.)

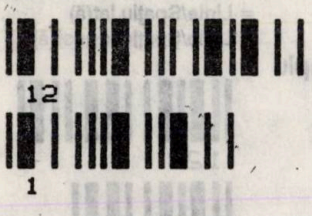
Tabela de coduri:

Caracterul	L1	S1	L2	S2	L3	S3	L4
1	0	0	0	0	1	1	0
2	0	0	0	1	0	0	1
3	1	1	0	0	0	0	0
4	0	0	1	0	0	1	0
5	1	0	0	0	0	1	0

6	0	1	0	0	0	0	1	
7	0	1	0	0	1	0	0	
8	0	1	1	0	0	0	0	
9	1	0	0	1	0	0	0	
0	0	0	0	0	0	1	1	
-	0	0	0	1	1	0	0	
$	0	0	1	1	0	0	0	
:	1	0	0	0	1	0	1	
/	1	0	1	0	0	0	1	
.	1	0	1	0	1	0	0	
+	0	0	1	0	1	0	1	
A	0	0	1	1	0	1	0	
B	0	1	0	1	0	0	1	
C	0	0	0	1	0	1	1	
D	0	0	0	1	1	1	0	

L1-L4 = Linia 1 - 4, S1-S3 = Spaţiul 1 - 3
1 = Linie/Spaţiu lat(ă)
0 = Linie/Spaţiu îngust(ă)

Exemplu



Cod 39

Generalităţi - cod alfanumeric. Pot fi reprezentate cifrele 0-9, 26 de litere, 7 caractere speciale. Fiecare caracter constă din 9 elemente (5 linii şi 4 spaţii). 3 elemente sînt late şi 6 sînt înguste, cu excepţia caracterelor speciale. Interspaţiul nu poartă informaţie.

Raportul de imprimare (R) linie îngustă: linie lată = 1:2 pînă la 1:3; spaţiu îngust: spaţiu lat = 1:2 pînă la 1:3. Dacă elementul îngust e mai mic decît 0,5 mm, atunci element îngust/element lat = 1:2,25

Avantaje: reprezentare alfanumerică

Dezavantaje: densitate mică de informaţie (4,8 mm/cifră la o lăţime a modulului de x = 0,3 mm şi R = 1:3.) toleranţă mică (10%).

Tabela de coduri:

Caracterul	L1	S1	L2	S2	L3	S3	L4	S4	L5
1	1	0	0	1	0	0	0	0	1
2	0	0	1	1	0	0	0	0	1
3	1	0	1	1	0	0	0	0	0
4	0	0	0	1	1	0	0	0	1
5	1	0	0	1	1	0	0	0	0
6	0	0	1	1	1	0	0	0	0
7	0	0	0	1	0	0	1	0	1
8	1	0	0	1	0	0	1	0	0
9	0	0	1	1	0	0	1	0	0
0	0	0	0	1	1	0	1	0	0
A	1	0	0	0	0	1	0	0	1
B	0	0	1	0	0	1	0	0	1
C	1	0	1	0	0	1	0	0	0
D	0	0	0	0	1	1	0	0	1
E	1	0	0	0	1	1	0	0	0
F	0	0	1	0	1	1	0	0	0
G	0	0	0	0	0	1	1	0	1
H	1	0	0	0	0	1	1	0	0
I	0	0	1	0	0	1	1	0	0
J	0	0	0	0	1	1	1	0	0
$	0	1	0	1	0	1	0	0	0
/	1	1	0	1	0	0	0	1	0
K	1	0	0	0	0	0	0	1	1
L	0	0	1	0	0	0	0	1	1
M	1	0	1	0	0	0	0	1	0
N	0	0	0	0	1	0	0	1	1
O	1	0	0	0	1	0	0	1	0
P	0	0	1	0	1	0	0	1	0
Q	0	0	0	0	0	0	1	1	1
R	1	0	0	0	0	0	1	1	0
S	0	0	1	0	0	0	1	1	0
T	0	0	0	0	1	0	1	1	0
U	1	1	0	0	0	0	0	0	1
V	0	1	1	0	0	0	0	0	1
W	1	1	1	0	0	0	0	0	0
X	0	1	0	0	1	0	0	0	1
Y	1	1	0	0	1	0	0	0	0
Z	0	1	1	0	1	0	0	0	0
-	0	1	0	0	0	0	1	0	1
.	1	1	0	0	0	0	1	0	0
BLANC	0	1	1	0	0	0	1	0	0
+	0	1	0	0	0	1	0	1	0
%	0	0	0	1	0	1	0	1	0
start/stop,0	0	1	0	0	1	0	1	0	0

L1-L5 = Linia 1 - 5, S1-S4 = Spaţiul 1 - 4
1 = Linie/Spaţiu lat(ă)
0 = Linie/Spaţiu îngust(ă)

Exemplu:

Cod EAN

Generalităţi - cod numeric, pot fi reprezentate cifrele 0-9. Fiecare caracter constă din 11 elemente. Toate liniile şi spaţiile poartă informaţie. Pot fi reprezentate numai grupuri de 8 sau 13 caractere împreună.

Avantaje - densitate mare de informaţie în 10 dimensiuni diferite.

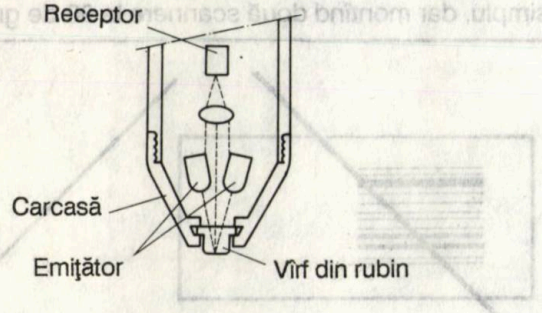
Dezavantaje - toleranţe foarte mici.

Tabela de coduri este prea complicată pentru a mai fi exemplificată acum. Ea poate fi solicitată de la Centrale für Coorganisation, Postfach 19 04 24, Spichernstrasse 55, 5000 Köln 1.

Tehnici de citire

Cititoare statice

Creioanele de citire funcţionează pe baza efectului ilustrat în figură. Emiţătoarele emit lumină roşie (630 mm) sau infraroşie (950 mm); creionul este "dus" deasupra hîrtiei, atingînd-o în permanenţă. Suprafaţa ce conţine codul de bare e luminată difuz. Prin intermediul unui sistem optic complex, lumina reflectată ajunge la fototranzistor. La parcurgerea codului de bare, sistemul transformă optic liniile şi spaţiile într-un tren de impulsuri. Acesta este disponibilizat în formă numerică la ieşire. Un decodor ataşat transformă trenul de impulsuri în date pe care le transmite serial (printr-o interfaţă RS232, RS422 sau 20mA)

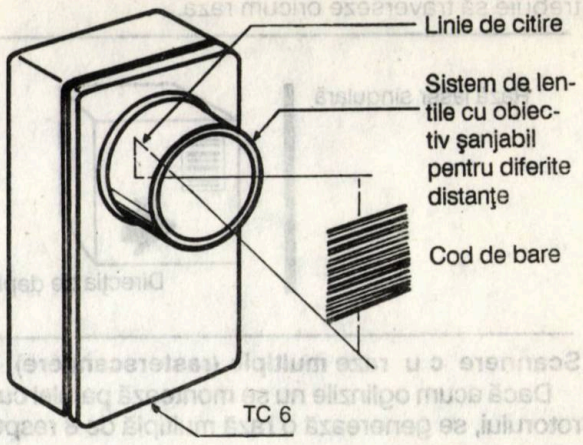


Cititoarele distanţoare folosesc acelaşi sitem, cu diferenţa că citirea se face la o distanţă de 10 mm +- 1 mm. Înălţimea minimă a codului de bare este de 3mm.

Cititoare dinamice

Sistemul cu cameră de luat vederi

Un rînd de fotodiode este dispus în spatele focarului, ca la o cameră suprafaţa filmului. Prin interogare internă rapidă a punctelor individuale, un cod de bare este transformat astfel într-un rastru de puncte. Cel puţin 4 pixeli trebuie să fie proiecţia unei linii înguste pentru ca aceasta să poată fi recunoscută. Utilizînd obiective variabile, pot fi recunoscute coduri de bară la pînă 4 m distanţă. Ca funcţie, camera corespunde unui scanner cu rază singulară. Numărul de citiri pe secundă (scans) este de pînă în 1000.



Scannere cu He - Ne

Este cunoscută reflexia unei raze de lumină printr-o oglindă netedă, la care unghiul de reflexie coincide cu unghiul de incidenţă.

Dacă 4 sau mai multe oglinzi se montează pe un rotor şi se reflectă razele luminoase ale soarelui sau ale unei surse de lumină artificiale atunci pe o suprafaţă se proiectează un punct care se deplasează pe un traseu rectiliniu, repetitiv. Dacă rotorul se învîrte suficient de repede, atunci funcţie de turaţia lui şi numărul de oglinzi, punctul luminos face pînă la 800 de parcurgeri ale traseului pe secundă (scans) pe suprafaţa pe care se proiectează.

De ce laser cu He - Ne?

Laserul are proprietatea de a emite lumină aproape paralelă. Astfel e posibilă reflexia ordonată a luminii cu un efort optic rezonabil (în principal, elemente optice netede, oglinzi). Prin paralelismul dat, citirea se poate face cu multă exactitate în profunzime.

Scanner cu rază singulară

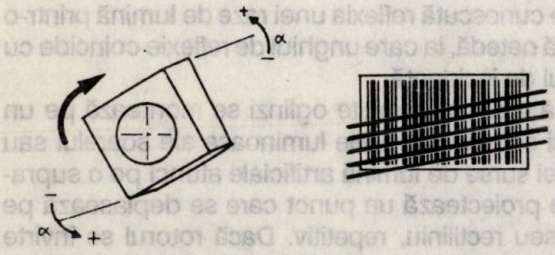
Folosind ca sursă de lumină un laser cu He - Ne, se ajunge la principiul de funcţionare al unui scanner cu laser He - Ne. Raza de lumină generată are un diametru de 0,2 - 0,6 mm. Dacă oglinzile se montează apoi paralel cu axa de rotaţie a rotorului, se proiectează o rază singulară de lumină.

Acest scanner se foloseşte cînd pe o bandă transportoare liniile de cod sînt paralele cu direcţia deplasării iar direcţia citirii este în unghi drept faţă de aceasta. Liniile sînt orientate orizontal, iar raza laser vertical. Avantajul este că nu trebuie realizată o precizie prea mare a înălţimii la care se plasează codul (+- 600 mm).

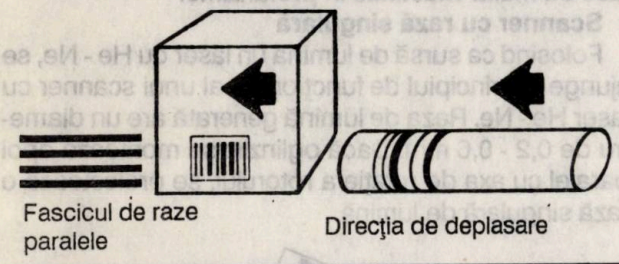
În afară de aceasta, prin deplasare, codul de bare trebuie să traverseze oricum raza.

Scannere c u raze multiple (rasterscannere)

Dacă acum oglinzile nu se montează paralel cu axa rotorului, se generează o rază multiplă de 8 respectiv 10 raze - rastrul.



Rasterscanner-ul se foloseşte atunci cînd liniile codului pot fi aplicate numai vertical faţă de direcţia de deplasare a încărcăturii (imprimanta poate tipări numai într-o direcţie, eticheta poate fi aplicată numai într-o anumită poziţie, etc.). Astfel se asigură o anumită toleranţă în ce priveşte înălţimea codurilor.



Scanner evantai

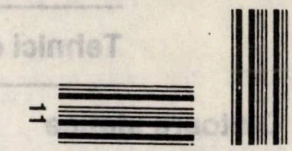
Prin plasarea unei oglinzi cu înclinare oscilantă în faţa unui scanner cu rază singulară, se poate citi un cod de bare de pe o suprafaţă, ca şi în cazul rasterscanner-ului. Corespunzător, distanţele dintre razele individuale depind de viteza de înclinare şi de amplitudinea oscilaţiilor

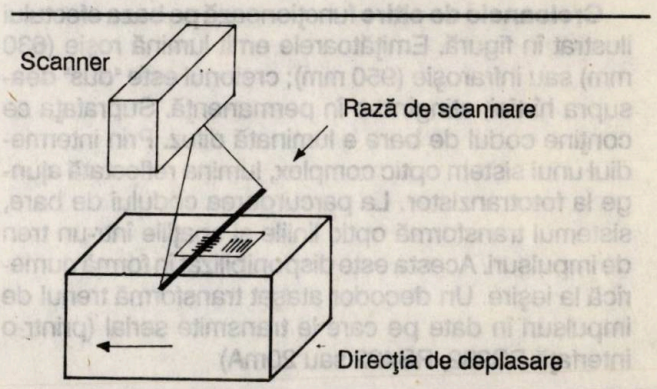
Citirea omnidirecţională

Dacă lucrurile ce se identifică nu au o orientare totdeauna aceeaşi, folosind un cod adecvat, acesta poate fi citit omnidirecţional.

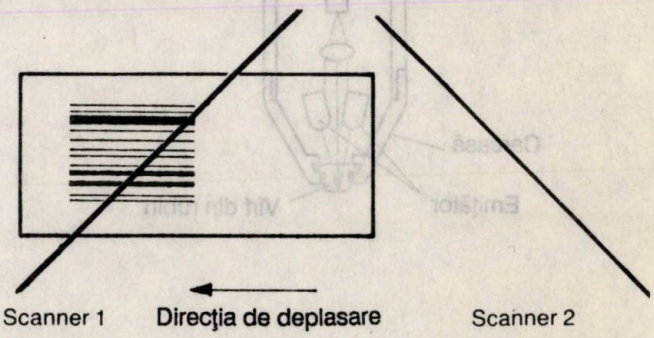
Eticheta în T

Acest cod se aplică de două ori, sub un unghi de 90. Premisa necesară este ca înălţimea codului (lungimea liniilor) să fie mai mare decît lăţimea.



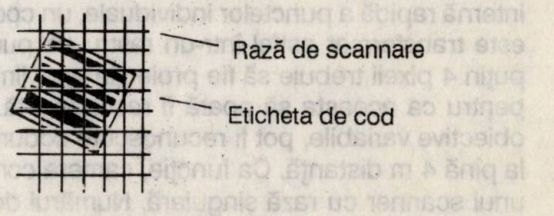


Citirea omnidirecţională cu două scannere cu rază individuală este o altă posibilitate, folosind un cod simplu, dar montînd două scannere la 90 de grade.

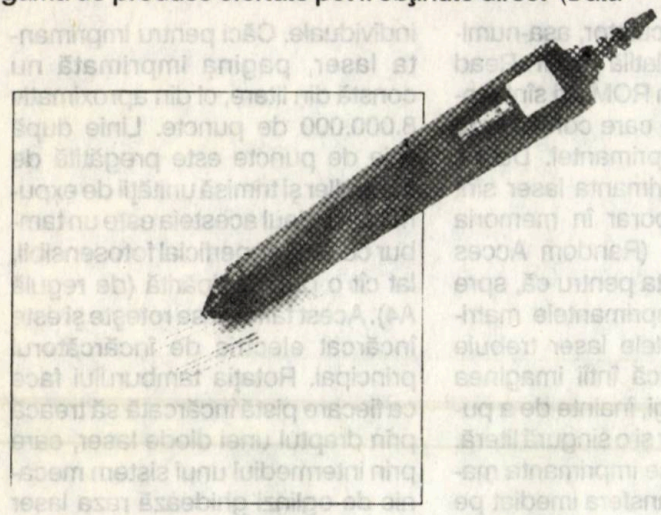


Scannere omnidirecţionale

Cu un scanner care nu emite numai unul sau mai multe raze paralele, ci o reţea completă, pot fi citite apoi toate codurile care au o construcţie cel puţin pătratică (lungimea liniilor = lăţimea lor), indiferent de poziţia codului faţă de scanner. Cel mai adesea, astfel de sisteme se utilizează la casele de marcat. Din păcate, profunzimea în citire este limitată la circa 15 cm.



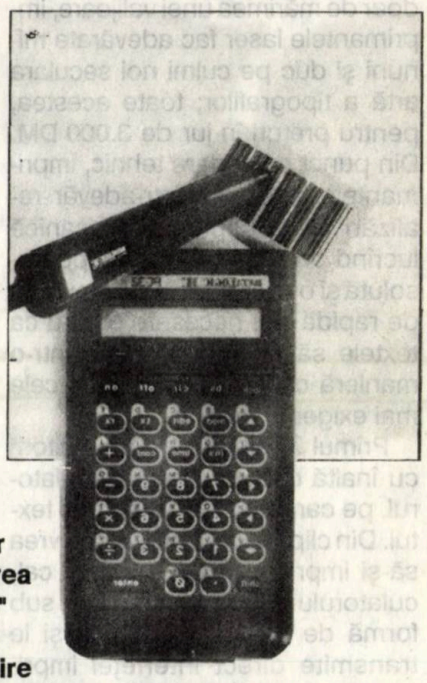
În final, vă prezentăm şi cîteva dispozitive, împreună cu aplicaţiile tipice. Ilustraţiile se bazează pe prospecte ale firmei Datalogic. Relaţii suplimentare relativ la gama de produse ofertate pot fi obţinute direct (Data-Logic International GmbH, Dreisteinstrasse 49 b, A - 2372 Giesshubel bei Wien, Austria). Altă firmă "cu nume" în domeniu: APOG, Paris, 298 av. G'ral de Gaulle, 92.140 Clamart, France.



Creion de citire

Aplicaţii:

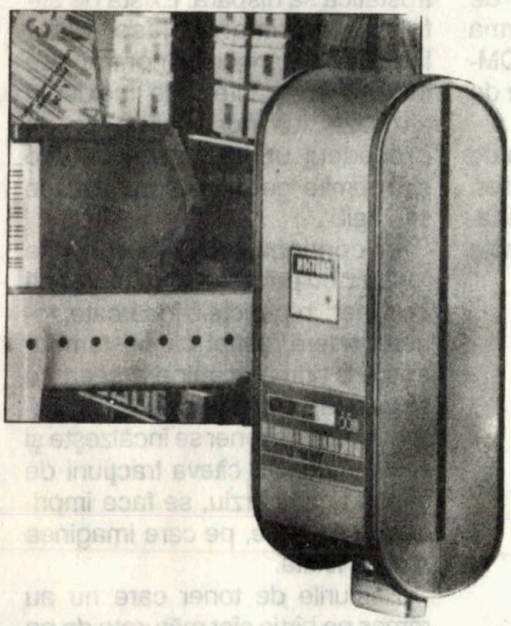
- case de marcaj în magazine cu autoservire
- inventarieri
- identificarea de pachete, paleţi, scrisori, acte, cărţi, fotografii, probe de date de producţie, administraţie, laboratoare



Microcalculator pentru introducerea datelor "pe teren" prin tastatură sau creion de citire

Aplicaţii:

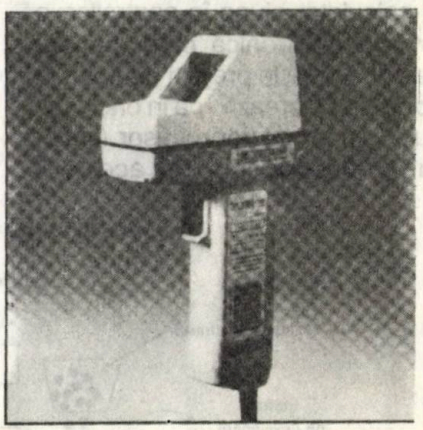
- preluarea şi memorarea datelor "pe teren" - inventare - consignaţie - preluare de comenzi - preluarea unor date de producţie - supravegerea producţiei - pontaj



Laser-scanner cu He-Ne

Aplicaţii:

- preluarea codurilor de bare fixate pe hîrtie, carton, folii sau etichete
- identificarea automată a bagajelor la aeroporturi
- identificare pachete, paleţi etc.



Pistol de citire cod de bare

Aplicaţii

- citirea fără atingere a diverse coduri de bare, în toate domeniile tehnicii şi comer1ului
- preluarea de coduri aplicate pe hîrtie, carton, etichete, suport textil, în poziţii oarecari
- casă şi preluare de marfă
- identificarea unor părţi de montaj pe linii de asamblare

Cum funcţionează imprimantele laser?

Gutenberg ar fi impresionat: doar de mărimea unei valijoare, imprimantele laser fac adevărate minuni şi duc pe culmi noi seculara artă a tipografilor; toate acestea, pentru preţuri în jur de 3.000 DM. Din punct de vedere tehnic, imprimantele laser sînt într-adevăr realizări de excepţie: o mecanică lucrînd cu o precizie aproape absolută şi o microelectronică extrem de rapidă sînt necesare pentru ca textele să ajungă pe hîrtie într-o manieră care să satisfacă şi cele mai exigente gusturi.

Primul în şiragul de "muncitori" cu înaltă calificare este calculatorul, pe care utilizatorul îşi scrie textul. Din clipa în care utilizatorul vrea să-şi imprime textul, interfaţa calculatorului preia aceste date sub formă de semnale electrice şi le transmite direct interfeţei imprimantei. Important e desigur ca cele două interfeţe să se înţeleagă şi să transmită (respectiv să preia) datele fie în şir indian, una cîte una ("serial"), fie în coloană de cîte opt deodată ("paralel").

Ajunse la imprimantă, datele de la calculator ajung în controller, o unitate electronică care prepară datele spre a le pregăti de tipar. În controller lucrează - ca în orice calculator - un microprocesor (creierul calculatorului). Pe acesta îl ajută, ca şi în calculator, aşa-numita memorie nevolatilă (ROM - Read Only Memory). În ROM-uri sînt stocate programele care controlează funcţionarea imprimantei. Datele preluate de imprimanta laser sînt depozitate temporar în memoria volatilă - RAM - (Random Acces Memory). Aceasta pentru că, spre deosebire de imprimantele matriceale, imprimantele laser trebuie să-şi construiască întîi imaginea unei pagini întregi, înainte de a putea imprima chiar şi o singură literă. Motivul: în timp ce imprimanta matriceală poate transfera imediat pe hîrtie semnalele de la calculator, deoarece lucrează : "rînd cu rînd", imprimanta laser imprimă numai pagini complete, pe care pune texte, imagini grafice sau ambele.

Deosebit de multă memorie în imprimantele laser necesită imaginile grafice, căci ele trebuiesc reprezentate punct cu punct în memorie. În contrast, literele, deşi pe hîrtie constau deasemenea din puncte individuale, au nevoie de foarte puţină memorie. Căci forma lor e deja preprogramată în ROM-uri, astfel că la nevoie se preia de acolo.

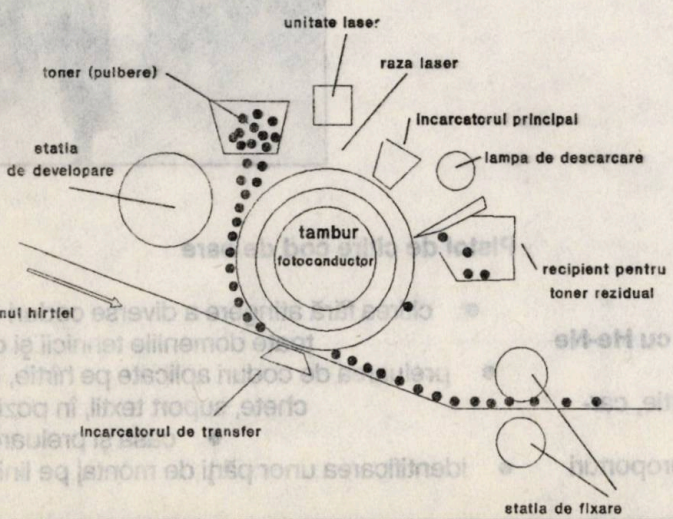
Din clipa în care toate datele de la calculator au ajuns la controller, acesta începe cu calculele necesare pentru a le transforma în puncte individuale. Căci pentru imprimanta laser, pagina imprimată nu constă din litere, ci din aproximativ 8.000.000 de puncte. Linie după linie de puncte este pregătită de controller şi trimisă unităţii de expunere. Nucleul acesteia este un tambur cu strat superficial fotosensibil, lat cît o pagină tipărită (de regulă A4). Acest tambur se roteşte şi este încărcat electric de încărcătorul principal. Rotaţia tamburului face ca fiecare pistă încărcată să treacă prin dreptul unei diode laser, care prin intermediul unui sistem mecanic de oglinzi ghidează raza laser astfel încît să ajungă pe pistă. Pentru a ocoli această mecanică complicată, unii producători folosesc diode luminescente (LED-uri) în locul laserului. Circa 2.000 de LED-uri sînt aşezate pe o "stinghie" una lîngă alta şi pot fi comandate individual.

Indiferent că e laser sau LED, lumina produsă face ca pe tambur, în locul luminat, încărcătura electrostatică să dispară. Există de altfel şi imprimante laser care lucrează pe principiul contrar - se încarcă în punctele de incidenţă ale razei luminoase. Funcţie de procedeul utilizat, imprimantele sînt numite "ce scriu negru" sau "ce scriu alb".

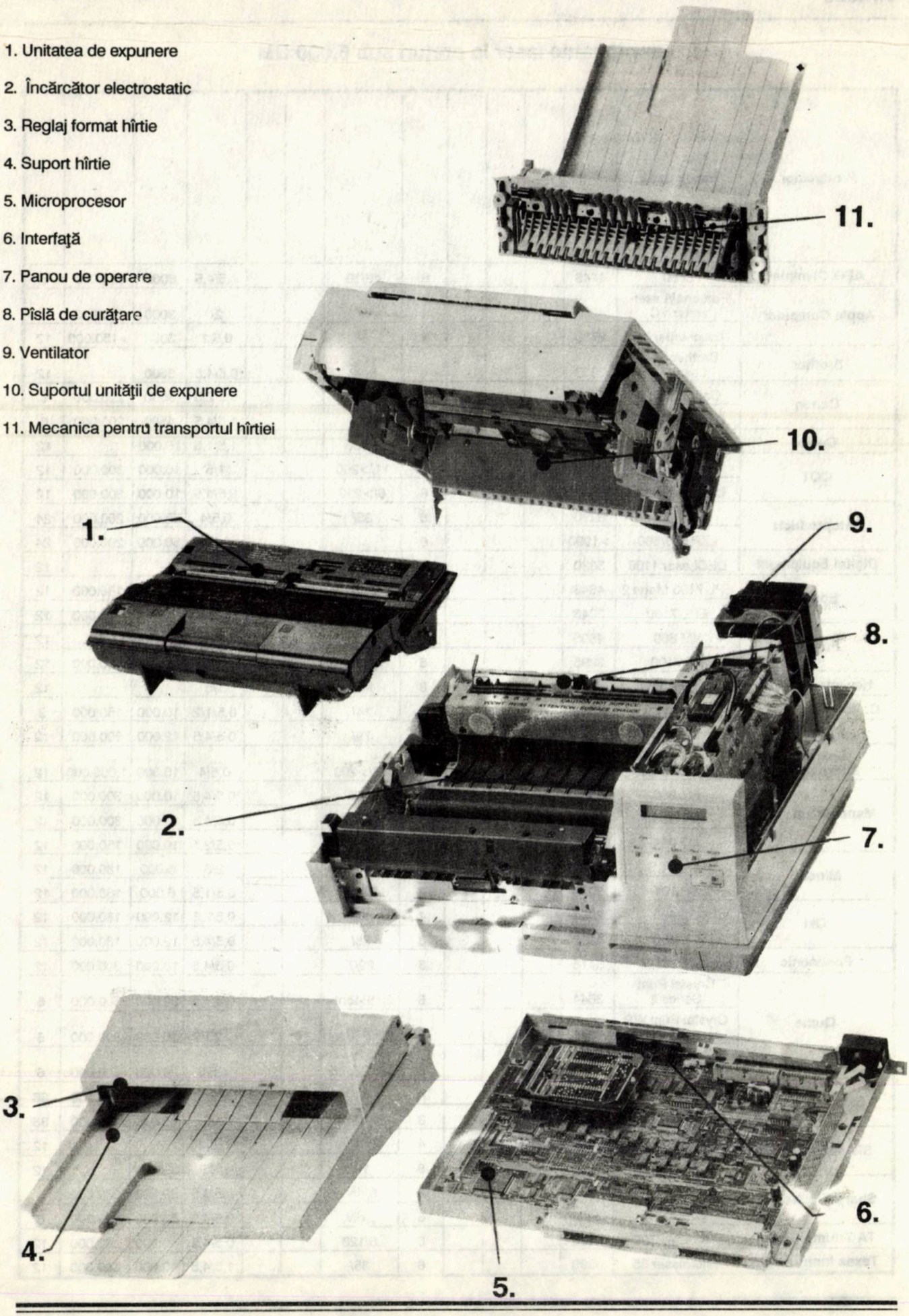
În continuare, tamburul este "prăfuit" cu pulberea extrem de fină de toner; în punctele încărcate, tonerul aderă, astfel că pe tambur apare o primă imagine a ceea ce se va tipări. Pentru aceasta, pulberea uscată de toner se încălzeşte şi se topeşte. Cu cîteva fracţiuni de secundă mai tîrziu, se face imprimarea pe hîrtie, pe care imaginea rămîne fixată.

Resturile de toner care nu au rămas pe hîrtie sînt măturate de pe tambur şi colectate de cîteva mici pensule înainte ca tamburul să fie descărcat. În staţia de fixare, tonerul şi hîrtia se răcesc, înainte ca o pagină imprimată să iasă, în sfîrşit, din imprimantă.

Principiul de funcţionare



1. Unitatea de expunere
2. Încărcător electrostatic
3. Reglaj format hîrtie
4. Suport hîrtie
5. Microprocesor
6. Interfaţă
7. Panou de operare
8. Pîslă de curăţare
9. Ventilator
10. Suportul unităţii de expunere
11. Mecanica pentru transportul hîrtiei

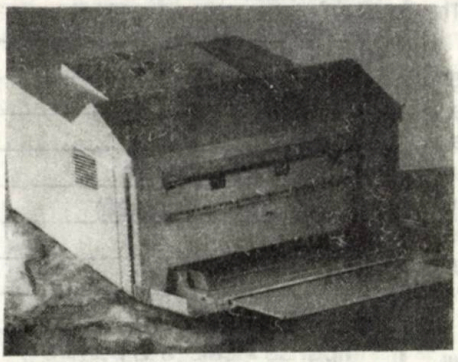


Imprimante laser la preţuri sub 5.000 DM

Producător	Imprimantă												
AEG Olympia	LP60	4446				6	49/10			0,5/4,5	6000	180.000	6
Apple Computer	Personal/Laser-writer SC	4350				4	4/			2/	3000	150.000	12
	Laserwriter LS	2730				4	4/			0,5/1	3000	>150.000	12
Brother	Brother HL-4 Laser AS	3933				4	5/22			0,5/4,5	3500		12
Canon	Canon LBP-4	3450				4	4/19			0,5/2,5	3500	200.000	6
	Canon LBP-8III	4950				8	4/19			1,5/4,5	4500	200.000	6
Casio	LCS-1600	1190				6	13/60			1,5/1,5	10.000		12
COT	Laserpro 90/8	4902				8	11/>200			1/5	10.000	300.000	12
	Laserpro EXEX	3409				6	6/>200			0,5/4,5	10.000	300.000	12
Dataproducts	LZR 650	2290				6	35/			0,5/4	50.000	200.000	24
	LZR 650/660	>1990				6				0,5/4	50.000	200.000	24
Digital Equipment	DECLaser 1100	3830				4	36/			0,5/2,5			12
Epson	EPL-7100 Mega 2	4648				6	11/13			2/6	6.000	180.000	12
	EPL-7100	3948				6	11/13			0,5/6	6.000	180.000	12
Fujitsu	VM 800	4695				8	7/8			1/5	8.000		12
	RX 7100	3995				5	6/30			0,64/4	6.000	180.000	12
Hewlett-Packard	HP Laserjet III	4400				8	14/8			1/5			12
C. Itoh Electronics	CI-4	3306				4	14/			0,5/1/2	10.000	150.000	9
Kyocera	F 800 T	4218				8	79/			0,5/4,5	12.000	300.000	12
Lexmark International	IBM 4019-E01	4611				5	10/>200			0,5/4	15.000	1.000.000	12
Mannesman Tally	MT 906	4503				6	6/35			0,5/4,5	10.000	300.000	12
	MT 905	4503				6	6/div.			0,5/4,5	10.000	300.000	12
	MT 904	3352				4	14/35			0,5/2,5	10.000	150.000	12
Minolta	SP 101 S	4891				6	48/			2/3	6.000	180.000	12
	SP 101	3979				6	32/44			0,5/1,5	6.000	180.000	12
Oki	OL 400	2998				4	15/			0,5/2,5	12.000	180.000	12
	OL 800	4198				8	25/			0,5/4,5	12.000	180.000	12
Panasonic	KX-P 4420	3418				8	22/7			0,5/4,5	18.000	300.000	12
Qume	Crystal Print Series II	3644				6	2/HP-fonts			0,5/1,5	20.000	300.000	6
	Crystal Print WP Plus	2845				6	2/HP-fonts			0,25/6	20.000	300.000	6
	Crystal Print Super Series II	4555				6	7/HP-fonts			1,5/6	20.000	300.000	6
Sanyo Electric	SPX 508	3180				8	9/128			0,5/5	15.000	300.000	36
	SPX 608	3408				8	9/128			1/5	15.000	400.000	36
Siemens Nixdorf	9021	3400				4	14/			0,5/4			12
	4810 P	4900				8	14/			1/5			12
Sharp Electronics	JX-9500H	3933				9	6/div.			0,5/4,5	50.000	200.000	6
	JX-9500 E	3408				6	6/div.			0,5/4,5	50.000	200.000	6
TA Triumph-Adler	TA SDR 7706	4320				6	8/128			0,5/4,5	10.000	300.000	12
Texas Instruments	Microlaser 35	4990				6	35/			1,5/4,5	40.000	300.000	12

Imprimanta HP Laserjet IIIP

Noua imprimantă Laserjet IIIP ce va înlocui imprimantele de tip HP Laserjet IIP, îmbină tehnologia seriei III cu preţul avantajos al predecesoarelor sale.



Cu Laserjet IIISi, Hewlett-Packard şi-a delimitat poziţia în segmentul superior al seriei III. Cu noul Laserjet IIIP, producătorul îşi rotunjeşte seria şi în jos. La preţul de vînzare de 3500 DM, Hewlett-Packard doreşte să-şi accentueze poziţia sa de lider al pieţii şi în segmentul preţurilor coborîte.

Cu excepţia efigiei, noul Laserjet IIIP nu se deosebeşte cu nimic, ca aspect exterior, de predecesorul lui IIP. Toate detaliile carcasei au rămas aceleaşi. Şi panoul operator - care în practică s-a dovedit reuşit - a fost preluat întocmai.

Ocupînd o suprafaţă de 34x63 cm (inclusiv caseta de hîrtie), Laserjet IIIP are loc aproape oriunde. Înălţimea este de 21 cm; cu caseta, opţională, de hîrtie, ce poate fi amplasată sub imprimantă, se mai adaugă 6 cm.

În mod normal, alimentarea cu hîrtie se face din faţă, folosind o casetă de alimentare multifuncţională, ce are o capacitate de 70 de pagini şi se potriveşte şi pentru plicuri. Deşi Laserjet IIIP nu este concepută ca o imprimantă de mare performanţă, ci ca o imprimantă cu volum mediu de tipărire, rezerva de 70 de pagini este mai degrabă insuficientă. Caseta opţională de hîrtie, ce poate fi amplasată sub imprimantă, rezolvă două probleme deodată. Pe de o parte nu mai este necesară caseta obişnuită de alimentare cu hîrtie, suprafaţa ocupată reducîndu-se astfel la 34x40,5 cm. Pe de altă parte rezerva de hîrtie creşte la 250 de pagini. Mecanica pretenţioasă utilizată face ca această casetă opţională să fie suficient de scumpă: 400 DM.

Multe detalii constructive evidenţiază faptul că IIIP este un produs matur, bine conceput. Astfel, de ex., mai multe cîrlige mici de ridicare, amplasate pe partea superioară a dispozitivului de antrenare a hîrtiei, împiedică paginile care ies dim imprimantă să mai antreneze în mişcare paginile deja tipărite.

Instalarea casetei de toner, care conţine toate materialele consumabile, se face prin faţă. Caseta este surprinzător de mică şi poate fi montată dintr-o singură mişcare. Ea este suficientă pentru cca. 3500 de pagini şi costă cca. 240 DM. Astfel o pagină ajunge la un cost de 6,9 pfennigi (fără costul echipamentului), respectiv 7,6 pfennigi /pagină (inclusiv echipamentul); ceea ce face ca IIIP-ul să se situeze în domeniul cel mai coborît de preţuri din categoria imprimantelor ce nu sînt capabile de Postscript.

Legarea imprimantei se poate face fie serial, fie paralel. Ambele interfeţe fac parte din configuraţia standard. Tot în configuraţia standard - lucru neobişnuit la imprimante din această categorie de preţuri - imprimanta este echipată cu 1 MByte de memorie; cu module corespunzătoare, memoria poate fi extinsă pînă la 4 MByte.

Imprimanta Laserjet IIIP este complet compatibilă cu Laserjet III şi, ca toate imprimantele seriei III, dispune de noul limbaj de comandă pentru imprimante PCL 5.

Sprijinită de mai toate programele cunoscute (în mod special de Windows 3.0, cu toate aplicaţiile), imprimanta poate fi folosită la parametrii maximi fără drivere suplimentare.

Cele mai importante noutăţi la PCL 5, faţă de PCL 4, sînt fonturile scalabile , grafica vectorială, direcţia de imprimare, modelul de imprimare şi îndesirea rastrului.

În afara celor 14 fonturi fixe, IIIP oferă suplimentar 8 fonturi ce pot fi dimensionate, bazate pe "Intellifont Scaling Technology" de la Agfa Compugraphic. Pe baza conturului, aceste fonturi sînt convertite la dimensiuni între 0,25 şi 999,75 puncte. Cînd un program solicită o anumită dimensiune, aceasta este calculată şi pregătită în memoria imprimantei.

Grafica vectorială integrată în PCL 5 este limbajul grafic propriu Hewlett-Packard, HPGL/2. Prin intermediul acestuia IIIP poate fi adresat şi ca plotter. Funcţiile de grafică vectorială pot fi combinate cu celelalte funcţii din PCL 5.

Astfel, fonturile scalabile pot fi integrate într-o grafică vectorială şi pot fi imprimate într-o direcţie oarecare. Suplimentar, sînt disponibile efecte speciale, cum ar fi fonturi înclinate sau lărgite.

Dacă pînă acum totdeauna era problematic să reuşeşti scrieri în direcţii distincte pe aceeaşi pagină, acum acest lucru este uşor realizabil cu PCL 5. Orice text poate fi imprimat în patru direcţii, rotite cu cîte 90 grade.

Prin funcţia de "model de imprimare" se înţelege, de ex., posibilitatea ca obiectele să fie "umplute" cu tonuri de gri sau modele diferite şi de a suprapune imagini diferenţiat. Astfel, Laserjet IIIP dispune de tonuri de gri rezidente şi de modele de rastru, care duc la imprimări uniforme. În afară de aceasta, folosirea lor duce la o mărire a vitezei de lucru, deoarece imprimanta produce rastrul intern,

acesta nemaitrebuind "adus" bit cu bit prin interfaţă de la calculator. Prin posibilitatea suplimentară de a suprapune diverse elemente transparent sau opac, se pot obţine efecte interesante, ca scrieri umbrite sau reprezentări inverse.

La imagini scanate, de ex., îşi găseşte aplicaţie "îndesirea de rastru". Este vorba despre un procedeu de compresie care, aplicat fişierelor-imagini, economiseşte spaţiu pe harddisk. În afară de aceasta, se reduce timpul necesar transferului lor din calculator spre imprimantă.

Un alt avantaj al întregii familii Laserjet III este procedeul RET (Resolution Enhancement Technology), care duce la o calitate vizibil îmbunătăţită a imprimării. Un cip special, propriu HP, selectează şi prelucrează în timp real pachete de 6x2560 puncte, pentru a corecta, de ex., liniile oblice, curbele sau întretăierile de linii. Deşi rezoluţia la Laserjet IIIP este de doar 300 dpi (dot per inch), această prelucrare suplimentară duce la o imprimare vizibil mai bună, variind mărimea şi poziţia punctelor. Printr-o modulare a razei laser, dimensiunea punctului poate fi variată în 5 trepte. Chiar la liniile înclinate, binecunoscutul "efect treaptă" poate fi eliminat aproape complet. La întretăierile de linii, punctele de rastru mai mici evită obişnuitele îngroşări şi apariţia "bulgărilor" de toner.

Intensitatea procedurii RET poate fi variată în 3 trepte; sau poate fi invalidată complet. Un rastru special pe o imprimare de test face posibilă găsirea variantei optime.

Îmbunătăţirile se văd cel mai bine la compararea cu imprimarea fără RET. Avantajul este vizibil şi fără lupă şi "aruncă" Laserjet IIIP-ul mult în faţă în ceea ce priveşte calitatea imprimării, la nivelul celorlalte reprezentante ale seriei III.

În ceea ce priveşte viteza de imprimare, IIIP marchează progrese considerabile faţă de modelul II. Procesorul intern (un Motorola 68000) lucrează acum la 16 MHz (10 MHz la modelul II), iar viteza de transfer a crescut de 4 ori.

Şi limbajul de comandă al imprimantei a devenit mai rapid, lucru vizibil mai ales la grafică.

Fără îndoială, cu Laserjet IIIP, Hewlett-Packard confirmă poziţia sa de lider pe piaţa imprimantelor laser, echilibrînd acum şi sub aspectul preţului situaţia care avantaja, uneori, concurenţa.

(I.F.)

Caracteristici

Nume	**HP Laserjet IIIP**
Preţ (inclusiv MwSt.)	cca. 3500 DM
Carcasa	
Dimensiuni (L x Î x l în cm)	34 x 21 x 63
Date tehnice	
Tehnica imprimării	Laser
Unitatea de imprimare	Canon
Rezoluţia (text şi grafică)	300 x 300 dpi
Tambur	organic
CPU	Motorola 68000
Memoria (min./max.)	1 MByte / 5 MByte
Timpul de încălzire la pornire	45 sec.
Listări într-o lună pînă la	8000 pagini
Durata de viaţă	nelimitată
Interfeţe	
Paralelă	da
Serială	da
Apple Talk	opţional
Altele	RS242 opţional
Firmware	
Familii de fonturi	Courier, Line Printer, Times, Univers
Număr de fonturi	67
Emulări	HP Laserjet III
Opţiuni hîrtie	
Formate	DIN A4, C5
Greutatea (g/m^2)	60 pînă la 105
Rezerva	70 / 320 pagini
Antrenare	manuală / automată
Ieşire (face-down / face-up)	70 / 20 pagini
Materiale consumabile	
Toner	-
Tambur	-
Unitatea de încălzire	-
Cartuş toner/tambur	3500 pag., cca. 240 DM
Costuri	
Preţ pe pagină (cu / fără preţul aparatului)	7,61 / 6,87 pfenning
Altele	
Garanţie	12 luni
Accesorii	Program hardcopy
Manuale	în engleză

Specificatia Lotus/Intel/Microsoft/Ast (XMS) pentru memoria extinsa

Specificatiile L/I/M/A pentru memoria extinsa (numite si specificaţiile XMS) cuprind, printre altele, unele tehnici de gestiune a memoriei al căror scop principal este acela de a furniza un standard pentru diverşi producatori de software care intentionează utilizarea memoriei extinse. Aceasta, la prima ei apariţie - pe PC AT - nu era însoţită de nici o metodă care să permită partajarea ei de mai multe aplicaţii; această regretabilă absenţă a dus la apariţia mai multor metode de gestiune incompatibile între ele, precum desuetul procedeu VDISK.

Chiar dacă sînt prezente într-un fişier care se încarcă ca un "device driver" normal, funcţiile XMS nu sînt apelate prin metodele tipice driver-elor DOS, ci invocînd Int 2fH cu coduri de funcţie specifice; datorită acestui fapt "driver"-ul XMS este denumit manager de memorie extinsă (eXtended Memory Manager).

Datorită unei scăpări în proiectarea microprocesorului 80286 există un mod REAL de adresare directă a unei părţi din memoria aflată deasupra primului megabyte, care teoretic ar trebui să fie inaccesibilă. După cum este ştiut, adresele utilizate de 8086 sînt codificate pe 20 biţi, fiind compuse dintr-un segment şi un deplasament (offset). În exemplul următor acestea se combină astfel:

Offset		9	9	9	9	+
Segment		F	E	D	C	=
Adresa	0	8	7	5	9	

După cum se observă 8086 ignoră transportul pe bitul 20, rezultînd astfel o referinţă la adresa 08759H. 80286 foloseşte adrese pe 24 biţi şi nu ignoră transportul obţinînd o referinţă la adresa 108759H, deasupra primului megabyte.

Pentru a obţine compatibilitatea cu programe care au fost scrise pentru 8086 (şi care în unele cazuri se bazau în mod explicit pe adresarea cu pierdere de transport), IBM a realizat AT-ul cu linia A20 a magistralei de adrese permanent dezactivată, ignorînd astfel transportul generat de 80286. Linia poate fi totuşi activată la cererea programelor aplicative, care în acest caz pot adresa direct primii 64 KB ai memoriei extinse punînd în segment valoarea FFFFH; valorile acoperite pornesc de la FFFFH:0010H la FFFFH:FFFFH, adică 64 KB - 16 bytes de memorie extinsă.

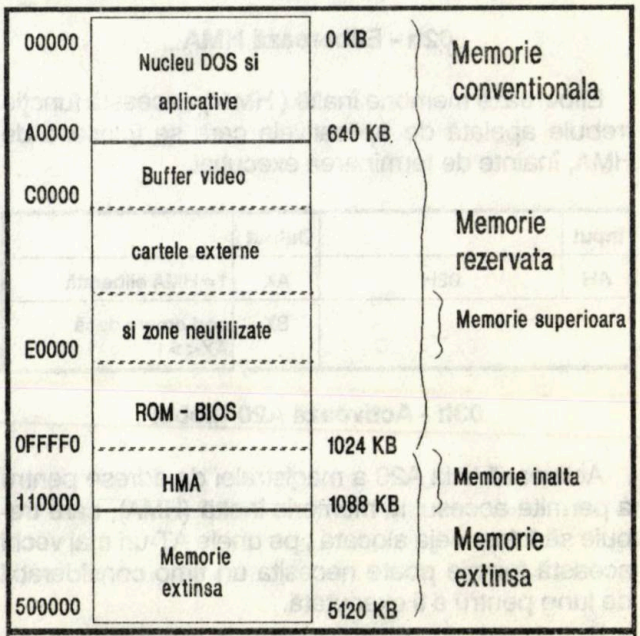
Termeni noi introduşi de specificaţiile XMS - L/I/M/A:

- **Memorie superioară** (Upper Memory) - cu acest termen se desemnează zona de memorie rezervată pentru a avea acces la alte tipuri de memorie prin intermediul unor driver-e şi a unei serii de adrese altfel neutlizate.
- **Memorie înaltă** (HMA - High Memory Area) - memorie deasupra primului megabyte adresabilă direct cu microprocesorul 80286 în mod real; nu poate fi mai lungă de circa 64 KB .

Pentru a permite accesul la memoria înaltă, ea trebuie disponibilizată prin încărcarea driver-ului XMS precizat în CONFIG.SYS. Avantaje:

- - utilizatorul poate decide dacă să instaleze sau nu funcţiile XMS şi dacă să permită sau nu utilizarea memoriei înalte;
- - utilizatorul poate asigura instalarea funcţiilor înaintea oricărui program care le-ar putea apela;
- - producătorii pot realiza "device driver"-e diferite.

Specificaţiile XMS - L/I/M/A au fost furnizate pentru prima dată de Microsoft printr-un driver numit HIMEM.SYS .



Organizarea memoriei conform specificaţiilor XMS

Sumarul funcţiilor XMS

Funcţiile XMS se pot obţine apelînd o rutină a cărei adresă este furnizată în urma unui apel la o funcţie INT 2Fh, după cum se va ilustra mai departe. Transmiterea parametrilor, rezultatelor se face prin intermediul regiştrilor, specific pentru fiecare funcţie în parte.

00h - Citeşte numărul versiunii XMS

Restituie numărul de versiune al managerului de memorie extinsă XMS.

Input		Output	
AH	00H	AX	Numărul vers. BCD 0123H=1.23
		BX	Versiune internă XMS
		DX	1=Există HMA

01h - Cere acces la HMA

Cere accesul la memoria înaltă (HMA) şi o alocă dacă aceasta e disponibilă şi dacă dimensiunea cerută este mai mare sau egală cu parametrul /HMAMIN (din linia de comandă device = himem.sys /HMAMIN ...).

Input		Output	
AH	01H	AX	1=HMA alocată
DX	Dimensiunea cerută FFFFH pentru aplicative liberă pentru TSR	BL	cod de eroare dacă AX <> 1

02h - Eliberează HMA

Eliberează memoria înaltă (HMA) ; această funcţie trebuie apelată de aplicativele care se folosesc de HMA, înainte de terminarea execuţiei.

Input		Output	
AH	02H	AX	1=HMA eliberată
		BX	cod eroare dacă AX<>1

03h - Activează A20 global

Activează linia A20 a magistralei de adrese pentru a permite accesul la memoria înaltă (HMA), care trebuie să fi fost deja alocată ; pe unele AT-uri mai vechi această funcţie poate necesita un timp considerabil de lung pentru a fi executată.

Input		Output	
AH	03H	AX	1=A20 activată
		BL	cod eroare dacă AX<>1

04h - Dezactivează A20 global

Dezactivează linia A20 a magistralei de adrese reducînd accesul la modul 8086, în care adresa FFFFh:0010h coincide cu 0000h:0000h; trebuie să fie apelată de programe care au controlul HMA şi au activat linia A20, înainte de terminarea execuţiei; execuţia ei poate necesita un timp considerabil.

Input		Output	
AH	04H	AX	1=A20 dezactivată
		BL	cod eroare dacă AX<>1

05h - Activează A20 local

Activează linia A20 a magistralei de date pentru a permite accesul la memoria extinsă; execuţia ei poate necesita un timp considerabil.

Input		Output	
AH	05H	AX	1=A20 activată
		BL	cod eroare dacă AX<>1

06h - Dezactivează A20 local

Dezactivează linia A20 a magistralei de date, reducînd accesul la memorie la modul 8086, În care adresa FFFFh:0010h coincide cu 0000h:0000h; această funcţie trebuie să fie apelată de programe care să fi activat anterior linia A20 înainte de terminarea lor; execuţia ei poate necesita un timp considerabil.

Input		Output	
AH	06H	AX	1=A20 dezactivată
		BL	cod eroare dacă AX<>1

07h - Cere starea liniei A20

Verifică dacă linia A20 este corect activată sau dezactivată.

Input		Output	
AH	07H	AX	1=A20 activată

08h - Cere dimensiunea memoriei extinse libere

Returnează dimensiunea celui mai mare bloc de memorie extinsă disponibil precum şi memoria extinsă liberă totală, excluzînd memoria înaltă dacă aceasta a fost deja alocată.

Input		Output	
AH	08H	AX	Dimensiune bloc în KB
		DX	Memorie liberă totală în KB
		BL	Cod eroare

09h - Alocă un bloc de memorie extinsă

Alocă un bloc de memorie extinsă şi restituie identificatorul unui handle XMS, dacă există memorie disponibilă suficientă; de notat că numărul disponibil de handle-uri XMS este limitat, acestea trebuind cerute cu moderaţie.

Input		Output	
AH	09H	AX	1= Bloc alocat
DX	Dimensiunea cerută în KB	DX	Handle XMS
		BL	Cod eroare dacă AX<>1

0Ah - Eliberează un bloc de memorie extinsă

Eliberează un bloc de memorie extinsă alocat anterior; această funcţie trebuie apelată înainte de terminarea programului, pentru a evita ca blocul respectiv de memorie să rămînă permanent alocat şi inutilizabil de către alte aplicţii.

Input		Output	
AH	0AH	AX	1= Bloc eliberat
DX	Handle XMS alocat anterior	BL	Cod eroare dacă AX<>1

0Bh - Mută un bloc de memorie extinsă

Deplasează un bloc de memorie extinsă între două zone de memorie extinsă sau convenţională; linia A20 nu trebuie să fie activată în timpul acestui apel dacă memoria înaltă este implicată în deplasare; pentru a indica o adresă de memorie convenţională, fie sursă fie destinaţie se atribuie valoarea zero handle-lui iar adresa în forma segment:offset se atribuie deplasamentului.

Input		Output	
AH	0BH	AX	1= Bloc mutat
DS:SI	Adresa buffer parametri	BL	Cod eroare dacă AX<>1

formatul buffer-ului cu parametrii de transfer:

DD lungimea ce urmează a fi transferată, în bytes
DW handle-ul blocului sursă
DD deplasament în interiorul blocului sursă, în bytes
DW handle-ul blocului destinaţie
DD deplasament în interiorul blocului destinaţie, în bytes

0C h- Protejează un bloc de memorie extinsă

Împiedică deplasările în cadrul blocului şi returnează adresa liniară de 32 biţi a unui bloc anterior alocat; această funcţie este util să fie apelată înainte de comutarea în modul protejat de adresare al procesorului.

Input		Output	
AH	0CH	AX	1= succes
DX	Handle XMS anterior alocat	DX:BX	Adresă liniară de 32 biţi
		BL	Cod eroare dacă AX<>1

0Dh-Deblochează un bloc de memorie extinsă

Deblochează un bloc de memorie anterior protejat; această funcţie trebuie utilizată înainte de terminarea programelor care au blocat porţiuni de memorie extinsă.

Input		Output	
AH	0DH	AX	1= succes
DX	Handle XMS anterior alocat	BL	Cod eroare dacă AX<>1

0Eh- Citeşte informaţii despre un handle XMS

Furnizează informaţii relativ la un bloc anterior alocat de memorie extinsă.

Input		Output	
AH	0EH	AX	1= succes
DX	Handle XMS anterior alocat	BH	1= bloc de memorie protejată
		BL	Număr handle-le disponibile sau Cod eroare dacă AX<>1
		DX	Lungime bloc în KB

0Fh - Realocă un bloc de memorie extinsă

Modifică dimensiunea unui bloc de memorie extinsă neprotejată.

Input		Output	
AH	0FH	AX	1= bloc realocat
DX	Handle XMS	BL	Cod eroare dacă AX<>1
BX	Noua dimensiune în KB.		

10h - Cere un bloc de memorie superioară

Cere accesul la un bloc de memorie superioară aliniat la o adresă de paragraf (segment)

Input		Output	
AH	10H	AX	1= bloc alocat
DX	Dimensiune în paragrafe	BX	Segment de început al blocului (de 16 biţi) alocat
		BL	Cod eroare dacă AX<>1
		DX	Lungimea în paragrafe a blocului alocat

11h - Eliberează un bloc de memorie superioară

Eliberează accesul la un bloc de memorie superioară; această funcţie trebuie apelată înainte de terminarea programelor care au alocat blocuri de memorie superioară.

Input		Output	
AH	11H	AX	1= bloc eliberat
DX	Segment de început al blocului alocat	BL	Cod eroare dacă AX<>1

Verificarea prezenţei driver-ului XMS şi apelarea lui

Înainte ca o aplicaţie să apeleze funcţii XMS trebuie obligatoriu să verifice prezenţa driver-ului în memorie. Acesta se poate face conform următoarei scheme:

- Se utilizează INT 2Fh pentru a verifica prezenţa driver-ului XMS apelînd funcţia AX = 4300, care trebuie să returneze Al = 80h în cazul în care driverul a fost instalat corect iar accesul la memoria extinsă funcţionează corespunzător;
- Se utilizează INT 2Fh pentru a obţine adresa de intrare în driver-ul XMS, apelînd funcţia AX = 4310h; adresa este returnată în perechea de registre ES:BX .
- Driver-ul XMS se apelează prin transferul execuţiei la adresa obţinută prin tehnica anterioară, după ce în prealabil au fost pregătite registrele pentru transmiterea de parametri.

Sumarul codurilor de eroare

În continuare este prezentat tabelul tuturor codurilor de eroare XMS posibile, furnizate de funcţiile apelate totdeauna în registrul BL cînd AX = 0.

Cod	Descriere
00h	Nu este eroare ;
80h	Funcţie neimplementată ;
81h	Dispozitiv asemănător VDISK instalat ;
82h	Eroare în timpul tratării liniei A20 ;
8Eh	Eroare generică de driver ;
8Fh	Eroare fatală de driver ;
90h	Nu există HMA ;
91h	HMA deja alocată ;
92h	Dimensiune cerută inferioară parametrului /HMAMIN
93h	HMA nealocată
A0h	Nu există memorie extinsă liberă
A1h	Handle-le XMS epuizate
A2h	Handle XMS eronat
A3h	Handle XMS sursă eronat
A4h	Deplasament sursă eronat
A5h	Handle XMS destinaţie eronat
A6h	Deplasament destinaţie necorespunzător
A7h	Lungime incorectă
A8h	Deplasament cu depăşire
A9h	Eroare de paritate
AAh	Bloc neprotejat
ABh	Bloc protejat
ACh	Depăşire număr de blocuri permise
ADh	Protejare imposibilă
B0h	Memorie superioară insuficientă
B1h	Nu există memorie superioară disponibilă .
B2h	Segment eronat

Consideraţii pentru dezvoltarea de aplicaţii

Consideraţiile următoare ar trebui urmate în proiectarea oricărei aplicaţii care face apel la interfaţa XMS:

- transferurile de date în DMA nu sînt recomandate în memoria înaltă;
- nu se recomandă transferul ca argumente pentru INT 21h, INT 25h sau INT26h al adreselor din memoria înaltă;
- se recomandă ca vectorii de întrerupere să nu se direcţioneze în memoria înaltă, întrucît dacă linia A20 s-ar dezactiva ulterior, vectorii respectivi n-ar mai putea fi atinşi şi gestiunea întreruperilor s-ar pierde.

Pentru nefericiţii prizonieri ai modului real de adresare prezentăm în cele ce urmează o bibliotecă completă de funcţii XMS pentru Turbo Pascal 6.0., care ar putea fi un colac de salvare pînă la apariţia şi impunerea unui standard DPMI (Interfaţă cu modul protejat de adresare).

Biblioteca a fost testată cu Turbo Pascal 6.0 şi driver-ul XMS aparţinînd lui Microsoft - HIMEM.SYS ver 2.0 (care s-a dovedit a nu suporta toate funcţiile specificaţiei XMS: lipsesc funcţiile de acces la memoria superioară), pe următoarele configuraţii : 386 cu ISA şi 8 MByte RAM şi respectiv AT 286 cu 1 MByte RAM.

(ing. Flaidăr Alin , ing. Udrea Alin)

```
{$B-,D-,F-,L+,R-,S-,V-}

unit XMS;

{ Micro ATCI, mai 1991 }
interface

var XMSError: byte;

function XMSInstalled: boolean;

{ the following routines should be called only after a
    successful call to XMSInstalled
}
procedure XMSInfo( var Version, InternalVersion: word;
        var HMAavail: boolean);
    { major version in low byte, minor in high byte }
procedure AllocHMA( bytes: word );
procedure FreeHMA;

procedure EnableAddrA20;
procedure DisableAddrA20;
procedure EnableDataA20;
procedure DisableDataA20;
function A20Status: boolean;

procedure AllocExtMem( KBytes: word;
            var Handle: word );
procedure FreeExtMem( Handle: word );
procedure FreeExtMemInfo( var MaxFreeBlock,
    TotalFreeMem { in KBytes } : word );
procedure ReAllocExtMem( KBytes, handle: word );

procedure MoveExtMem(DestOffs
        DestHandle : word;
        SourceOffs  : longint;
        SourceHandle    : word;
        bytes       : longint);

procedure ReadExtMem( bytes   : longint;
        SourceHandle : word;
        SourceOffs   : longint;
        var Dest);

procedure WriteExtMem( var Src;
        bytes    : longint;
        DestHandle : word;
        DestOffs : longint);

procedure GetHandleInfo( handle
        var locked : boolean;
        var HLength { KBytes },
        FreeHandles: word);

function GetProtectedMemory( handle: word ): pointer; { 32 bits
    linear pointer}
procedure ReleaseProtectedMemory( handle: word );

procedure AllocSupMem( PagesRequested {16 bytes paragraphs}
:word;
    var PagesAllocated {16 bytes paragraphs}:word;
    var BaseAddress     :pointer );
procedure FreeSupMem( BaseAddress: pointer );

implementation
uses Dos;

var XMSAddr   : pointer;
    _AX,_BX,_DX,_SI : word ;
    var buff : record
    TransferBytes : longint;
    source  : word;
    SourceOffset : longint;
    destination : word;
    DestOffset : longint
end;

function XMSInstalled : boolean;

var regs: registers;
begin
    with regs do;
    begin
        AX := $4300;
        intr($2F, regs);
        if AL  $80 then XMSINstalled := false
        else
        begin
         AX := $4310;
         intr($2F, regs);
         XMSAddr := ptr( ES,BX );
         XMSInstalled := true
        end
    end
end;    { XMSInstalled }

procedure CallXMS;
begin
    asm
    mov AX,[_AX]
    mov BX,[_BX]
    mov DX,[_DX]
    mov SI,[_SI]
    call [XMSAddr]
    mov [_AX],AX
    mov [_BX],BX
    mov [_DX],DX
    end                 : longint;
end;    { call XMS }


procedure XMSInfo( var Version, InternalVersion: word;
    var HMAavail: boolean);
    { major version in low byte, minor in high byte }
begin
    _AX := 0;
    CallXMS;
    Version := _AX;
    InternalVersion := _BX;
    HMAavail := _DX = 1
end;

{
begin                   : word;
    asm
    mov AH,0
    call [XMSAddr]
    mov word [Version], AX
    mov word [InternalVersion], BX
    end
end;
}

procedure FreeExtMemInfo( var MaxFreeBlock,
    TotalFreeMem { in KBytes } : word );
begin
    _AX := $0800;
    CallXMS;
    XMSError := lo(_BX);
    if XMSError = 0 then
    begin
        MaxFreeBlock := _AX;
        TotalFreeMem := _DX
    end
end;    { FreeExtMem }

procedure AllocHMA( bytes: word );
begin
```

```
    _AX := $0100;
    _DX := bytes;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { AllocHMA }

procedure FreeHMA;
begin
    _AX := $0200;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { FreeHMA }

procedure EnableAddrA20;
begin
    _AX := $0300;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { EnableAddrA20 }

procedure DisableAddrA20;
begin
    _AX := $0400;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { DisableAddrA20 }

procedure EnableDataA20;
begin
    _AX := $0500;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { EnableDataA20 }

procedure DisableDataA20;
begin
    _AX := $0600;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { DisableDataA20 }

function A20Status: boolean;
    { true -- enabled; false -- disabled }
begin
    _AX := $0700;
    CallXMS;
    A20Status := _AX = 1
end;

procedure AllocExtMem( KBytes: word;
    var Handle: word );
begin
    _AX := $0900;
    _DX := KBytes;
    CallXMS;
    if _AX = 1 then
    begin
        XMSError := 0;
        handle := _DX
    end
    else XMSError := lo(_BX)
end;    { AllocExtMem }

procedure FreeExtMem( Handle: word );
begin
    _AX := $0A00;
    _DX := handle;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX);
end;

procedure MoveExtMem( DestOffs : longint;
    DestHandle : word;
    SourceOffs : longint;
    SourceHandle : word;
    bytes : longint    );
begin
    move( bytes, Buff, sizeof(Buff) );
    _AX := $0B00;
    _SI := ofs( Buff);
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;

procedure ReadExtMem( bytes : longint;
        SourceHandle : word;
        SourceOffs : longint;
        var Dest);
begin
    with Buff do
    begin
        TransferBytes := bytes;
        source        := SourceHandle;
        SourceOffset  := SourceOffs;
        destination   := 0;
        DestOffset    := longint( @Dest )
    end;
    _AX := $0B00;
    _SI := ofs( Buff);
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { ReadExtMem }

procedure WriteExtMem( var Src;
    bytes    : longint;
    DestHandle : word;
    DestOffs : longint);
begin
    with Buff do
    begin
        TransferBytes := bytes;
        source        := 0;
        SourceOffset  := longint( @Src );
        Destination   := DestHandle;
        DestOffset    := DestOffs
    end;
    _AX := $0B00;
    _SI := ofs( Buff );
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { WriteExtMem }


function GetProtectedMemory( handle: word ): pointer; { 32 bits
    linear pointer}

type SegOfs = record
    Ofs, Seg: word
    end;
var LinPtr: pointer;

begin
    _AX := $0C00;
    _DX := handle;
    CallXMS;
    if _AX = 1 then
    begin
    SegOfs(LinPtr).seg := _DX;
    SegOfs(LinPtr).ofs := _BX;
    XMSError := 0;
    end
    else
    begin
    LinPtr := nil;
    XMSError := lo(_BX)
    end;
    GetProtectedMemory := LinPtr
end;
```

```
procedure ReleaseProtectedMemory( handle: word );
begin
    _AX := $0D00;
    _DX := handle;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { ReleaseProtectedMemory

procedure GetHandleInfo( handle
    var locked : boolean;
    var HLength { KBytes },
    FreeHandles: word);
begin
    _AX := $0E00;
    _DX := handle;
    CallXMS;
    if _AX = 1 then
    begin
        locked      := hi(_BX) = 1;
        FreeHandles := lo(_BX);
        HLength     := _DX;
        XMSError    := 0
    end
    else XMSError := lo(_BX)
end;    { HandleInfo }

procedure ReAllocExtMem( KBytes, handle: word );
begin
    _AX := $0F00;
    _BX := KBytes;
    _DX := handle;
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { ReAllocExtMem    }


procedure AllocSupMem( PagesRequested {16 bytes paragraphs}
:word;                  }
    var PagesAllocated {16 bytes paragraphs}:word;
    var BaseAddress :pointer );
begin
    _AX := $1000;
    _DX := PagesRequested;
    CallXMS;
    if _AX = 1 then
    begin
        PagesAllocated := _DX;
        BaseAddress    := ptr(_BX, 0);
        XMSError       := 0
    end
    else XMSError := lo(_BX)
end;    {   AllocSupMem }

procedure FreeSupMem( BaseAddress: pointer );
begin
    _AX := $1100;
    _DX := seg(BaseAddress^);
    CallXMS;
    if _AX = 1 then XMSError := 0 else XMSError := lo(_BX)
end;    { FreeSupMem }
end.
```

```
{$A+,B-,D-,F-,G+,O-,R-,S-,V-,X-}
{$M 16384,0,655360}

program TestXMS;

{ Micro ATCI, mai 1991 }

uses XMS, crt;
type MaxArray = array[1..65519] of byte;

var
    Ver,IntVer :    word;
    HMAOk      :    boolean;

    MaxFree,
    MemFree    :    word;
    GetA20Status:   boolean;
                    { top of conventional memory: }
    HMATestArray:       MaxArray absolute $FFFF:$0010;

    handle     :    word;
    locked     :    boolean;
    HandleLength,
    FreeHandles:    word;

    i          :    word;
    j,N        :    longint;
    T          :    array[0..255] of longint; { 1 KB long }

    pages      :    word;
    BaseAddress:    pointer;
    type ST2 = string[2];

    function HexString(Number: byte): ST2;
    function Hex_Char(Number: Word): Char;
    begin
        if Number<10 then
        Hex_Char:=Char(Number+48)
        else
        Hex_Char:=Char(Number+55);
    end; { Function Hex_Char }

    Var
    S: ST2;
    begin
    S:='';
    S:=S+Hex_Char(Number shr 4);
    Number:=Number and $0F;
    S:=S+Hex_Char(Number);
    HexString:=S
    end; { Function Hex_String }

procedure Error;
var msg: string[40];
begin
    write('Error '+ HexString(XMSError),'; ');
    case XMSError of
        0   :   msg := 'No error.';
        $80 :   msg := 'Inexisting function.';
        $81 :   msg := 'VDISK like device present.';
        $82 :   msg := 'A20 handling error.';
        $8E :   msg := 'Internal Error.';
        $8F :   msg := 'Fatal internal error';
        $90 :   msg := 'No HMA.';
        $91 :   msg := 'HMA already allocated.';
        $92 :   msg := 'Cannot allocate less than /HMAMIN.';
        $93 :   msg := 'HMA not allocated.';
        $A0 :   msg := 'Extended memory totaly allocated.';
        $A1 :   msg := 'No more handles available.';
        $A2 :   msg := 'Bad XMS handle.';
        $A3 :   msg := 'Bad source handle.';
        $A4 :   msg := 'Bad source offset.';
        $A5 :   msg := 'Bad destination handle.';
        $A6 :   msg := 'Bad destination offset.';
        $A7 :   msg := 'Length not allowed.';
        $A8 :   msg := 'Bad length with overlapping.';
        $A9 :   msg := 'Parity error.';
        $AA :   msg := 'Unprotected block.';
        $AB :   msg := 'Protected block.';
        $AC :   msg := 'Too many blocks.';
        $AD :   msg := 'Cannot protect block.';
        $B0 :   msg := 'Not enough upper memory.';
```

```
        $B1 :   msg := 'Not available upper memory.';
        $B2 :   msg := 'Bad Segment address.';
    end;
    writeln;
    writeln( msg);
    if XMSError in [$8E,$8F] then halt(1);
end;

begin
    if XMSInstalled then
    begin
    XMSInfo( Ver, IntVer, HMAOk);
    writeln('Found XMS driver version ',HexString( hi(Ver)),'.',
                    HexString(lo(Ver)),
    '; internal version ',HexString( hi(IntVer)),'.',
                    HexString(lo(IntVer)) );
    write('High Memory Area ');
    if HMAOk then writeln(' available.') else writeln(' not available');

    FreeExtMemInfo( MaxFree, MemFree);
    writeln('Free extended memory: ',MemFree,' kbytes.');
    writeln('Maximum free block: ',MaxFree,' kbytes.');
    GetA20Status := A20Status;
    if GetA20Status then writeln (' A20 line enabled.')
    else writeln('A20 line disabled.');

    if HMAOk then
    begin
        AllocHMA( 50000);
        if XMSError > 0 then Error
        else
        begin
            writeln('HMA succesfully allocated. ');
            if not GetA20Status then
            begin
                write(' Enabling A20 address line. This may take
                            some time...');
                EnableAddrA20;
                if XMSError > 0 then Error
                else
                begin
                 writeln('OK.');
                 write(' Enabling A20 data line. This may take
                            some time...');
                 EnableDataA20;
                 if XMSError > 0 then Error
                 else
                 begin
                    writeln('OK.');
                    write('Testing HMA...');
                    for i := 1 to 65519 do
                        HMATestArray[i] := i mod 255;
                    i := 1;
                    while (HMATestArray[i] = i mod 255) and
                        (i <= 65519) do inc(i);
                    if i < 65519 then writeln('HMA Test failed.')
                        else writeln('OK.');
                    DisableAddrA20;
                    if XMSError > 0 then Error;
                    DisableDataA20;
                    if XMSError > 0 then Error
                 end
                end
            end;
            FreeHMA;
            if XMSError > 0 then Error
        end
    end;
    write('Allocating ',MaxFree,' kbytes of extended memory...');
    AllocExtMem(MaxFree, handle);
    if XMSError > 0 then Error
    else
    begin
        writeln('OK.');
        GetHandleInfo( handle, locked, HandleLength,
            FreeHandles);
        if XMSError >0 then Error
        else writeln('Handle ',handle,' allocated; length: ',
            HandleLength,
                    '; ', FreeHandles,' more free handles.');
        writeln('Testing ',MaxFree,' kbytes extended memory:');
        write('Writing ',1024,'
        N := 0;
        for i := 1 to MaxFree do
        begin
            gotoxy(9,WhereY); write(N shl 2 +1024);
            for j := 0 to 255 do T[j] := N + j;
            WriteExtMem(T,1024,handle,N shl 2 );
            inc(N,256);
            if XMSError > 0 then
            begin
                Error;
                i := MaxFree+2
            end
        end;
        if i <= succ(MaxFree) then
        begin
            Gotoxy( 1,WhereY);
            write('Reading ',1024,'
            N := 0;
            for i := 1 to MaxFree do
            begin
             gotoxy(9,WhereY); write(N shl 2 +1024);
             ReadExtMem(1024,handle,N shl 2, T);
             if XMSError > 0 then
             begin
                Error;
                i := MaxFree+2
             end
             else
             begin
                for j := 0 to 255 do
                if T[j] <> j+N then
                begin
                    writeln;
                    writeln('      Error testing extended memory.');
                    i := MaxFree+2
                end;
             inc(N,256);
             end;
             if i <= succ(MaxFree)
             begin
              writeln;
              writeln(' OK.')
             end
          end
    end;
    write('Deallocating ',MaxFree,' KBytes of extended memory... ');
    FreeExtMem(handle);
    if XMSError > 0 then Error else writeln('OK.');
    end
    else writeln('XMS driver not installed.');
end.
```

Cobol = Cobol ?

Din cel puţin două puncte de vedere, Cobol-ul are caracteristici singulare: pe de o parte, este, incontestabil, cel mai utilizat limbaj de programare în domeniul comercial, iar pe de altă parte, (surprinzător?) este totodată unul din cele mai contestate limbaje de programare.

Faptul că este utilizat intens în ramurile corespunzătoare, ar trebui să fie un argument puternic în favoarea lui. Atunci, de unde motivele de ceartă? Orice programator cu experienţă în COBOL cunoaşte desigur punctele slabe care pot servi ca şi contraargumente pentru folosirea COBOL-ului - le vom trece pe scurt în revistă, încercînd să arătăm că mai puţin limbajul ca atare cît alte circumstanţe sînt cauze ale disputei şi că avantajele precumpănesc cu mult.

Să începem cu lucruri elementare: manualele de referinţă sînt toate, în limba engleză. Mai mult: autorii acestora ţin la o completitudine deseori exagerată şi informaţii foarte concentrate, ceea ce, deseori, duce la nenumărate trimiteri. Exemple clare, care să demonstreze descrierile stufoase ale sintaxei în mod practic, lipsesc de obicei cu desăvîrşire.

Chiar dacă unele compilatoare COBOL sînt însoţite de complexe programe demonstrative - rămîne de văzut dacă găsiţi acele pasaje care sînt interesante pentru munca Dvs. Chiar dacă aţi studiat informatica, dacă e să vă obişnuiţi cu un nou compilator, ar trebui să prevedeţi o perioadă de experimentări destul de lungă şi mult timp "pierdut".

În afară de aceasta, deseori listinguri kilometrice cu "algoritmi" îmbîcsiţi şi total nestructuraţi după criteriile moderne, sperie cititorul. Critica aproape că nu are sens; pentru că acolo unde se programează strict în COBOL, generaţii elitiste ţin cu dinţii de tradiţii, iar cei care intră sînt nevoiţi să se supună cu stricteţe "standardelor" specifice întreprinderii. Pentru programe regîndite şi cu deschidere spre viitor - posibile şi în Cobol, în ciuda afirmaţiilor contrare! - pur şi simplu nu rămîne timp.

Aşa că se continuă în ritmul obişnuit. Asemănător microprocesoarelor 80386 şi 80486, care, în ciuda arhitecturii celei mai inteligente şi unor capacităţi aproape inepuizabile, sînt nevoite să se lupte cu aplicaţii DOS cu multiple limitări de memorie, şi în Cobol, în ciuda faptului că în permanenţă se înlocuiesc standarde vechi cu unele mai noi şi se implementează extensii de limbaj tot mai funcţionale, "bătrînele" programe, care, deşi ticsite cu GOTO-uri, au acea calitate de bază care pînă la urmă decide valoarea unui program - merg! - rămîn în uz şi trebuie întreţinute, pe parcursul a ani şi ani de zile de muncă migăloasă. Nu-i de mirare dacă un programator răsfăţat în C sau în Pascal nu găseşte nici o plăcere în Cobol şi refuză să-şi lase alterat modul de gîndire - structurat - de reguli de mult depăşite. N-ajută la nimic vocabularul simplu, foarte asemănător englezei uzuale, şi nici modul vizual foarte "curat" în care se prezintă programele Cobol.

În afară de diversele minusuri ce ţin de gusturi individuale, un factor important care explică antipatia ce i se arată este faptul că pentru studenţii informaticieni, Cobol-ul este materie obligatorie. Relativ la forma şi logica Cobol-ului se poate într-adevăr discuta, căci el este realmente un limbaj de programare orientat pe problemă şi ar trebui privit ca atare. A programa calcule ştiinţifice - în afara domeniului comercial! - sau aplicaţii grafice în Cobol este aproape imposibil. Funcţii trigonometrice sau alte funcţii asemănătoare nu există în vocabularul standard - dar este adevărat că de regulă nici nu sînt necesare în aplicaţii Cobol.

Dar să trecem şi la avantaje. Cînd este vorba de prelucrarea rapidă şi exactă pînă la cîteva poziţii după virgulă a unor fişiere extrem de mari, cu tabele multidimensionale şi indexate, Cobol-ul este una din puţinele unelte de programare ce garantează succesul. Un motiv important care a dus la apariţia Cobol-ului a fost cerinţa de a putea face cele mai variate selecţii în fişiere oarecare. De aceea, Cobol-ul dispune de un sofisticat mecanism de control pentru fişiere şi tabele, cu toate variantele de acces imaginabile, inclusiv chei multiple. Cîmpurile de date ale unui program Cobol sînt ierarhizate în trepte, de la 1 la 49. Astfel e posibilă accesarea unor grupe întregi de date, folosind un singur nume. Variabilele şi constantele se divid în alfabetice, alfanumerice, numerice şi editate numeric şi se detaliază pînă la ultimul octet. Numai definirea inteligentă a unei variabile vă poate deja economisi importante eforturi de programare:

77 LEI PIC Z,ZZZ,ZZ9.99 VALUE ZERO.

Variabila editată numeric "LEI", de nouă poziţii, corespunde nivelului 77, adică nici nu are alte date subordonate şi nici nu se subordonează vreunei alte grupe de date. La afişarea pe ecran sau la imprimare, zerourile premergătoare din poziţiile marcate prin "Z" vor fi înlocuite cu blancuri. LEI are două poziţii zecimale, e formatată în mii şi iniţializată cu valoarea 0. (Printr-o clauză în program, scrierea americană a numărului poate fi înlocuită cu scrierea europeană, cu virgulă pentru partea zecimală şi punct ca separator de mii).

Există şi variabile booleene. Ataşate nivelului "88", ele pot corespunde în tabele şi mai multor condiţii - deci unor mulţimi!

Cu puţine excepţii sintaxa instrucţiunilor este identică la aproape toate compilatoarele Cobol. Aproape toate compilatoarele se

```
1   |  IDENTIFICATION DIVISION.
2   |
3   |  program-id.    BENCH.
4   |  author. umj,tj.
5   |
6   |  ENVIRONMENT DIVISION
7   |
8   |  CONFIGURATION SECTION.
9   |
10  |  source-computer.     AT-DR-DOS.
11  |  object-computer.     AT-DR-DOS.
12  |
13  |  INPUT-OUTPUT SECTION.
14  |
15  |  FILE-CONTROL.
16  |
17  |      select File
18  |      assign to 'BENCH.DAT'
19  |      file status IOResult
20  |      organization indexed
21  |      record key DataKey
22  |      access dynamic.
23  |
24  |  DATA DIVISION.
25  |
26  |  FILE SECTION.
27  |
28  |  fd  File
29  |      label record standard.
30  |  01 DataRecord.
31  |      05 DataKey pic 99.
32  |      05 DataChr pic x occurs 79 times.
33  |
34  |  WORKING-STORAGE SECTION.
35  |
36  |  77  IOResult     pic xx.
37  |  77  ZPos         pic 99.
38  |  77  SPos         pic 99.
39  |  77  Tasta        pic x  value space.
40  |  77  Pi           pic s9v9(9)    value 3.141592654.
41  |  77  x            pic s9v9(10).
42  |  77  s            pic s9v9(10) value zero.
43  |  77  i            pic 9(5).
44  |  77  f            pic 9(18) value 1.
45  |
46  |  01 Proba pic x.
47  |      88 Eroare value 'l'.
48  |
49  |  01 Matrice.
50  |      05 Rlesire pic x occurs 79 times.
51  |
52  |  01 Sinus.
53  |      02 Rind occurs 25 times.
54  |          03 Coloana occurs 79 times.
55  |              04 Car pic x.
56  |
57  |  PROCEDURE DIVISION.
58  |
59  |  begin.
60  |      move spaces to Proba, Sinus.
61  |      move -3.14 to x.
62  |      display 'Putintica rabdare...'.
63  |      perform Domeniu until x 3.14.
64  |      open output File.
65  |      if IOResult not = zero
66  |          display '1. Eroare in fisier: ', IOResult
67  |          accept Tasta
68  |          move 'l' to Proba
69  |      else
70  |          move zero to ZPos
71  |          perform Memorare until ZPos = 25.
72  |      close File.
73  |      if not Eroare
74  |          open input File.
75  |      if IOResult not = zero
76  |          display '2. Eroare in fisier: ', IOResult
77  |          accept Tasta
78  |      else
79  |          move zero to ZPos
80  |          perform Reciteste until ZPos = 25.
81  |      close File.
82  |  end.
83  |      stop run.
84  |
85  |  Domeniu.
86  |      move zero to i.
87  |      perform Calcul until i = 10.
88  |      compute SPos rounded = x / Pi * 39 + 40.
89  |      compute ZPos rounded = s * 11 + 13.
90  |      move 'x' to Car (ZPos, SPos).
91  |      add 0.1 to x.
92  |      move 0 to s.
93  |      move 1 to f.
94  |  Calcul.
95  |      compute f = f * (i * 2 + 1).
96  |      compute s = s + ((-1) ** i) * (x ** (2 * i + 1)) /
97  |      compute f = f * (i * 2 + 2).
98  |      add 1 to i.
99  |  Memorare.
100 |      add 1 to ZPos.
101 |      move ZPos to DataKey.
102 |      move zero to SPos.
103 |      perform Aduna until SPos = 79.
104 |      write DataRecord invalid key
105 |          display 'Eroare de scriere: ', IOResult
106 |          accept Tasta.
107 |  Aduna.
108 |      add 1 to SPos.
109 |      move Car (ZPos, SPos) to DataChr (SPos).
110 |  Reciteste.
111 |      add 1 to ZPos.
112 |      move ZPos to DataKey.
113 |      read File invalid key
114 |          display 'Eroare de citire: ', IOResult
115 |          accept Tasta.
116 |      move zero to SPos.
117 |      perform Afisare until SPos = 79.
118 |      display Matrice.
119 |  Afisare.
120 |      add 1 to SPos.
121 |      mov DataChr (SPos) to Rlesire (SPos).
122 |  *END OF BENCH.COB
123 |
```

Figura 1.

bazează pe vocabularul exact definit prin standardul ANSI-85, care este realmente foarte cuprinzător şi acoperă mulţumitor cele mai importante variante de programare.

Cu toate acestea, la o privire atentă portabilitatea se dovedeşte limitată. Puteţi desigur porta fără probleme un program Cobol din DOS spre Unix sau OS/2, şi după poziţionarea corespunzătoare a comutatorilor de compilare şi a altor opţiuni, obţineţi, într-un timp rezonabil, o variantă care să meargă - totuşi, probleme de adaptare apar întotdeauna.

Chiar şi respectînd standardul pînă în ultimul amănunt, trebuie să vă aşteptaţi la lungi liste de erori la prima compilare.

De cele mai multe ori este însă vorba de erori în lanţ, care de regulă dispar dintr-o singură "lovitură" (dacă găsiţi rădăcina răului...) Cum am mai menţionat, acest lucru se poate deseori rezolva modificînd o opţiune de compilare - sau cu o redefinire a unei instrucţiuni Cobol speciale.

Un exemplu posibil - un programator liber profesionist întreţine pentru diverse firme programe în Cobol pe PC-uri. Firma X lucrează numai cu compilatorul Micro Focus, în timp ce firma Y s-a specializat în Realia-Cobol. Acasă preferă să lucreze în MS - Cobol, versiunea 2.1. Ocazional, prin intermediul unui cunoscut, are de lucru cu MS-Cobol, versiunea 3.0. Cînd a învăţat, a învăţat utilizînd VS-Cobol de la IBM. Programatorul unei alte firme, care lucra în RM-Cobol, a renunţat la servici, drept pentru care probabil vor apare comenzi şi în legătură cu acest compilator. Astfel că numai în mediul DOS, programatorul are de a face cu 6 compilatoare Cobol.

Programul BENCH.COB, scris pentru a permite o comparaţie între diversele compilatoare, calculează sinusul folosind cele 4 operaţii aritmetice elementare, face apoi două ridicări la putere, scrie rezultatele sub formă de tabel într-un fişier, reciteşte fişierul şi reprezintă pe ecran o curbă stilizată. Nu e vorba desigur de un program deosebit de util, cît de a folosi, instrucţiuni Cobol cît mai variate şi de a observa "reacţia" diferitelor compilatoare. Singurul aspect tehnic este

viteza de execuţie, care s-a cronometrat. În rest, interesează numai portabilitatea programului.

Dificultăţi fundamentale nu există.

Privind însă scurtimea programului, multele mici diferenţe se însumează neplăcut. Aplicaţiile profesionale în Cobol umplu de regulă biblioraturi întregi, motiv pentru care "micile" diferenţe, chiar dacă stau doar în amănunte "banale", pot duce la timpi de transpunere importanţi.

Compilarea cu MS-Cobol, v.2.1 duce cu promptitudine la prima eroare. Numele fişierului "BENCH.DAT" trebuie precizat prin "VALUE of FILE-ID" din "FILE SECTION" şi nu se poate face ca la aproape toate celelalte compilatoare prin "ASSIGN TO" din paragraful "FILE CONTROL". Ecranul e şters în mod automat. Compilarea se face prin comenzile menţionate în tabelul alăturat; timpul de execuţie al modului Runtime ISAM nu a fost luat în considerare.

La MS-Cobol 3.0, ecranul nu se mai şterge automat; programul funcţionează însă impecabil. La fel şi la Microfocus şi Realia Cobol. Acestea sînt, ambele, protejate prin "dongles", lucru mai puţin stînjenitor la Microfocus Cobol decît la Realia Cobol. "Dongle"-ul de la Realia este deranjant de mare: a trebuit reamplasat adaptorul VGA pentru a-i face loc. În afară de aceasta, acest dongle vrea neapărat ca imprimanta să fie legată la prima interfaţă paralelă şi să fie neapărat online; mai mult, nici nu se împacă cu alte "dongle"-uri (ca de exemplu cel de la Microfocus).

MS-Cobol 3.0 şi Microfocus Cobol sînt aproape 100% compatibile. Însă în loc de IXSIO, Microfocus la linkeditare are nevoie de biblioteca EXTFX.

Acu-Cobol, exact ca şi MS-Cobol v2.1, are nevoie de clauza "ASSIGN TO DISK". în loc de "VALUE OF FILE-ID...", Acu-Cobol acceptă însă numai "VALUE OF LABEL...". Ecranul nu e şters în mod automat.

Nici la MBP, ecranul nu e şters. Instrucţiunea ACCEPT produce automat un semnal sonor.

La RM-Cobol, puţin deranjant este mesajul de terminare, care determină o defilare în sus a întregului ecran.

Probleme ridică, în general, instrucţiunile ACCEPT şi DISPLAY. Deoarece aceste instrucţiuni sînt prea puţin standardizate, fiecare producător "condimentează" după gustul propriu aceste instrucţiuni. Plasarea cursorului într-un anumit punct poate provoca cele mai neplăcute surprize. De aceea, ca programator profesionist de Cobol, care este nevoit deseori să porteze aplicaţiile scrise, ar trebui să renunţaţi din principiu la atributele ACCEPT şi DISPLAY specifice unui producător şi să utilizaţi în schimb interfaţa prin CALL. Prin aceasta, producătorii de compilatoare oferă posibilităţi de programare deosebit de individualizate - astfel, rutinele confortabile de tratare a ecranului rămîn locale şi se pot adapta independent de restul programului, "dintr-o singură mişcare".

(I.F.)

Cobol	Comenzile necesare lansării programului din fig.1	Timpul consumat (secunde)	Preţ (USA, apr. '91)
MS-Cobol 2.1	cobol bench isam runcob bench isam/f	34,6	
MS-Cobol 3.0	cobol bench.cob link bench ixsio bench	8,3	629 $
MS-Cobol 4.0			1299,60 DM (RFG, iunie '91)
Microfocus Cobol	cobol bench.cob link bench extfx bench	8,4	1499 $ (W/tools)
Realia Cobol	realcob bench link bench,,,realcob bench	6,6	844 $
Acu-Cobol		29,8	
MBP	viscob bench link vcmain bench,bench,nul,vcobol+vcsort+vcisam bench	12,5	
RM-Cobol	rmcobol bench runcobol bench	14,5	

Figura 2.

Un limbaj gigant cu un nou aspect

MS - Cobol V 4.0 PDS

Cel ce-şi aduce aminte de primul compilator Cobol pentru PC va rămîne perplex în faţa celei mai noi versiuni. Editoarele confortabile şi multele programe utile uşurează foarte mult munca programatorului profesional. În sfîrşit te poţi concentra asupra problemei principale - programarea. Multe munci suplimentare care înainte cădeau în sarcina programatorului sînt preluate acum de compilator.

Microsoft a realizat cu această ultimă versiune Cobol un pas consecvent în direcţia corectă. Fiind echipată cu o multitudine de instrumente de programare utilitare, versiunea 4.0 îşi merită pe deplin numele de PDS: Professional Development System.

Cel mai bătător la ochi este noul mediu de proiectare, cunoscut deja de la aite limbaje, şi anume PWB (Programmer's Workbench). Acesta garantează fără probleme trecerea de la un limbaj de programare la altul. Şi cel care lucrează doar în Cobol se va obişnui foarte repede cu noul mediu de editare. Acesta este realizat în standard SAA şi se poate deservi cu mouse-ul. Cel care renunţă la mouse poate lucra şi cu tastatura. Deservirea tastelor este standardizată.

Toate operaţiunile, care sînt necesare fiecărui program al pachetului se pot utiliza prin suprafaţa utilizator. La acestea se adaugă şi instalarea propriu-zisă a compilatorului, a linkeditor-ului şi a fiecărui debugger.

Există posibilitatea, foarte avantajoasă, de a deschide foarte multe ferestre în acelaşi timp. Acestea pot conţine de pildă surse program, texte de "help" sau mesaje de eroare.

Dacă totuşi lipseşte din PWB o funcţie des folosită, atunci ea se poate îngloba ulterior în mod foarte simplu. În acest scop trebuie scris doar un macrou corespunzător, care trebuie introdus în fişierul TOOLS.INI - Dacă este necesar, atunci un macrou propriu poate recurge la funcţii deja existente în PWS sau la alte macrouri.

Programatorii ştiu să aprecieze proprietatea de memorare a compilatorului la terminarea lucrului. La reluarea unui proiect de program se poate continua cu aceleaşi condiţii.

Help-ul on line este exemplar, el înlocuieşte căutarea repetată în manualul de utilizare. Dacă cursorul este plasat pe un cuvînt cheie Cobol, la apărarea tastei F1 apare o nouă fereastră. Această fereastră conţine toate informaţiile importante cu explicaţii aferente funcţiunii şi o descriere detaliată a sintaxei.

În domeniul detectării erorilor, Cobol 4.0 oferă multe instrumente de ajutor. "Source Browser" vizualizează informaţii importante despre procedurile existente şi despre legăturile variabilelor introduse. Singura condiţie este folosirea comutatorului corespunzător care duce la crearea unui fişier BSC ("Source Browser

```
$SET ANS85 NOOSVS MF
IDENTIFICATION DIVISION.
PROGRAM-ID. RECUR.
* Acest program demonstreaza utilizarea
* apelurilor recursive in MS-Cobol 4.0
AUTHOR. D-S.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01      tab1    pic x(02) occurs 10.
01      tab2    pic x(02) occurs 10.
LOCAL-STORAGE SECTION.
01      z1      pic 9(02).
LINKAGE SECTION.
01      z2      pic 9(02).
PROCEDURE DIVISION.
MAIN SECTION.
CALL X"E4".
MOVE 1 TO Z1
CALL "UMPLE" USING Z1.
MOVE 1 TO Z1.
CALL "TABELA" USING Z1.
STOP RUN.

UMPLE-TABELA SECTION.
ENTRY "TABELA" USING Z2.
IF Z2 > 10 THEN
DISPLAY "Tabela - 2 umpluta" at 1110
ELSE
MOVE TAB1 (Z2) TO TAB2 (Z2)
MOVE Z2 TO Z1
ADD 1 TO Z1
CALL "TABELA" USING Z1
END-IF.
EXIT PROGRAM.

INITIALIZARE-TABELA SECTION.
ENTRY "UMPLE" USING Z2.
IF Z2 > 10 THEN
DISPLAY "Tabela initializata" at 1010
ELSE
MOVE Z2 TO TAB1 (Z2) Z1
MOVE Z2 TO Z1
ADD 1 TO Z1
CALL "UMPLE" USING Z1
END-IF.
EXIT PROGRAM.

* RECUR.CBL  Sfirsit
```

Exemplu de apel recursiv în Cobol 4.0

Cache"). Acest fişier conţine toate informaţiile necesare ulterior lui Source Browser.

Dacă eroarea nu poate fi depistată astfel, atunci se vor folosi Code view - Debugger (doar sub OS/2) sau Animator - Debbuger. El poate fi iniţiat direct de pe suprafaţa utilizator dar el nu poate fi mînuit prin mouse.

Nou în comparaţie cu versiunea anterioară este şi generatorul de măşti integrat (screen-uri). Astfel măştile pentru programare se pot crea mult mai uşor. Nici generatorul de măşti nu poate fi deservit cu mouse.

Compilatorul generează un cod program mult mai compact şi mai simplu decît în versiunea precedentă. Aceasta se poate remarca mai ales la importul şi exportul de imagini (ecrane) şi la utilizarea fişierelor. Timpul de răspuns a scăzut în cazul I/E ecran cu 50%.

La fel de simplă ca elaborarea aplicaţiilor DOS este şi elaborarea aplicaţiilor pentru managerul de prezentare sub OS/2. Programatorul poate alege între o linkeditare statistică şi una dinamică a bibliotecilor sau a subprogramelor. Acest lucru este valabil şi pentru DOS.

Avantajele sistemului OS/2 sînt clare. Linkeditarea dinamică face ca toate subprogramele linkeditate corespunzător să devină părţi ale DLL (Dynamic Link Library) a OS/2. Asfel fişierele executabile obţinute sînt mult mai compacte şi mai rapide, deoarece părţile necesare sînt activate doar în timpul execuţiei programului. Astfel se face şi o economie serioasă de memorie.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. MAUS.
* Acest program demonstreaza modul
* de initializare al unui mouse
AUTHOR. D-S.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01      MOUSE-FUNCTIE PIC 9(02) COMP-X VALUE 64.
01      MOUSE-PARAM   PIC 9(02) COMP-X.
01      MOUSE-DETAILS-G PIC 9(02) COMP-X VALUE 67.
01      MOUSE-DETAILS.
05      MOUSE-X-POZ     PIC 9(04) COMP-X.
05      MOUSE-Y-POZ     PIC 9(04) COMP-X.
05      MOUSE-STATUS    PIC 9(04) COMP-X.
01      CHAR    PIC X.

PROCEDURE DIVISION.
CALL X"E4".
* Activare mouse
MOVE 1 TO MOUSE-PARAM.
CALL X"AF" USING MOUSE-FUNCTIE MOUSE-PARAM.
ACCEPT CHAR AT 1010.
* Dezactivare mouse
MOVE 0 TO MOUSE-PARAM.
CALL X"AF" USING MOUSE-FUNCTIE MOUSE-PARAM.
STOP RUN.

* MOUSE.CBL Sfirsit
```

Iniţializarea mouse-ului în Cobol

Pentru simplificarea programării de aplicaţii pentru Presentation Manager există programul suplimentar H2CPY. Acesta traduce fişierele de headere C în echivalentele lor Cobol. De multe ori merită încercată transformarea unui program C în unul Cobol. Astfel se poate economisi mult efort de scriere.

Totodată Cobol 4.0 oferă şi posibilitatea lucrului cu SQL. Arhitectura client-server a fost îmbogăţită în noua variantă cu posibilitatea de comunicare cu eventuali serveri de baze de date SQL. Compilatorul permite integrarea instrucţiunilor SQL în programele Cobol cît şi interogarea directă a bazelor de date SQL.

Pentru orice proiect suprafaţa utilizator generează o bază de date proprie proiectului, iar Cobolul memorează şi administrează toate informaţiile importante despre proiectul respectiv.

Această bază de date poate fi apoi deservită cu ajutorul lui Source Browser. Astfel manipularea programelor Cobol deosebit de voluminoase devine mult mai simplă. Mai ales Source Browser ajută programatorul să se descurce foarte repede într-un program Cobol necunoscut.

Programul răspunzător pentru aceasta se numeşte NMAKE. El recunoaşte imediat ce părţi de programe dintr-un anumit proiect au rămas nemodificate şi se ocupă doar de părţile care au suferit o modificare de la ultima prelucrare. O altă versiune specială a acestui program, denumită NMK reuşeşte să genereze şi programe voluminoase, care necesită mai multă memorie decît memoria disponibilă. Acest lucru are sens cînd se lucrează în Real Mode sau sub OS/2.

Programatorului i se oferă şi funcţiuni noi pentru programarea mouse-ului. Modul de iniţializare a mouse-ului ni-l arată programul MOUSE.CBL. Un dezavantaj îl constituie faptul că în manual se fac mereu trimiteri de la un capitol la altul şi faptul că exemplele de programare nu sînt grupate. Nu este chiar simplu, deci, să te orientezi, cu toate că în rest manualele nu lasă aproape nici o dorinţă neîndeplinită. În manuale există chiar şi o descriere a modului în care trebuie făcut transferul programelor sursă de pe sistemele IBM/370 pe PC-uri (Compatibility Guide).

Noi utilitare: ILINK - un linkeditor incremental, QH - poate vizualiza chiar şi helpuri online la nivelul liniei de comandă. Sistemul de HELP-uri este foarte bine construit, texte de help existente se pot modifica cu ajutorul utilitorului HELPMAKE, putînd fi adăugate şi texte noi.

(I.M.)

Încă o dată mai rapid

Foxpro 2.0

În cursa bazelor de date, Foxpro 2.0 a luat din nou un avans în faţa concurenţilor dBase IV şi Paradox 3.5.

Sistemul de gestiune al bazelor de date (SGBD) Foxpro 2.0 conţine o serie de îmbunătăţiri, comparativ cu versiunile anterioare, care ne îndreptăţesc să afirmăm că nu este vorba doar despre o actualizare cu modificări pur cosmetice. În primul rînd ne referim la viteza de lucru, care a fost sporită din nou. Memoria de lucru necesară a fost minimizată şi au fost adăugate facilităţi SQL, noi tipuri de indexuri şi utilitare pentru programatorii de aplicaţii.

Foxpro 2.0 va fi disponibil în două versiuni, care vor fi livrate însă împreună fără ca pentru aceasta să se perceapă o taxă suplimentară: o versiune pentru PC/AT şi o versiune extinsă (Extended Edition) pentru procesoare 80386. Utilizarea versiunii extinse este recomandată în cazul bazelor de date cu mai mult de 500.000 de articole, pentru a îmbunătăţi sensibil performanţele încă o dată. Cu aceasta Foxpro 2.0 nu numai că doboară două muşte cu o singură lovitură, dar foloseşte şi puternica arhitectură a magistralei de 32 de biţi.

În pas cu moda în sectorul bazelor de date şi Foxpro 2.0 lucrează acum cu instrucţiuni SQL (Structured Query Language). Foxpro 2.0 include instrucţiunea "Select" şi a fost extins şi cu alte instrucţiuni.

Interogările Select se autooptimizează în funcţie de dimensiunea tabelelor, indexurile disponibile, mărimea memoriei de lucru şi a spaţiului disponibil pe harddisk etc.

Comanda poate crea indexuri temporare pentru a mări performanţele. Acest SGBD poate fi astfel recomandat şi pentru utilizare în cazul lucrului în reţea.

Pe lîngă interogările SQL, Foxpro 2.0 dispune şi de "RQBE" (Relational Query By Example), o metodă puternică de interogare a bazelor de date prin exemplu. Chiar şi interogările SQL pot fi testate astfel comod de către începători şi poate fi învăţat astfel modul de comportare al unor comenzi.

Sistemul de management al bazei de date mai aduce o noutate care vorbeşte de la sine despre inventivitatea programatorilor de la Foxsoftware: tehnologia Rushmore. Este vorba despre o procedură de optimizare a interogării unei baze da date, dezvoltată în propria casă. După datele furnizate de Foxsoftware, această versiune atinge viteze de interogare de pînă la 1.000 de ori mai mari decît în cazul versiunilor anterioare, datorită acestei tehnici. Chiar şi bazele de date uriaşe cu milioane de articole vor putea fi prelucrate astfel pe PC-uri la viteze comparabile cu cele de pe mainframe-uri (calculatoarele mari). Avantajele tehnologiei Rushmore devin evidente abia în cazul utilizării versiunii extinse pentru volume uriaşe de date.

Foxpro 2.0 cunoaşte acum trei tipuri de indexuri: indexul standard (.IDX) al versiunilor Foxbare şi Foxpro 1.XX, noul "Compound Index" (.CDX) analog indexului "Multiple Index File" din dBase şi indexul compact. Fişierele de indexuri conţin mai mulţi pointeri. Suplimentar există un aşa-numit "structural compound index file", care este deschis automat în momentul deschiderii bazei de date complementare. În cazul indexului compact este vorba despre o formă comprimată a indexurilor sus-numite. Foxpro 2.0 utilizează, în acest caz, o tehnică proprie de comprimare care reduce fişierele index la cca. o şesime din dimensiunea lor iniţială. Chiar şi acest lucru măreşte viteza de execuţie, deoarece fişierele care trebuiesc citite de pe harddisk sînt mai mici iar decomprimarea se face în memorie.

Per ansamblu acest program lucrează cu o viteză care-i poate face invidioşi pe concurenţi ca dBase IV sau Paradox 3.5. Pentru acei care pun preţ pe viteza de răspuns la interogarea unei baze de date acesta ar putea fi un criteriu decisiv.

Foxpro s-a orientat pînă în prezent pe standardul dBase căruia a încercat să-i dea o putere sporită. Din acest motiv este asigurată compatibilitatea nu numai cu versiunile anterioare de Foxpro ci şi cu dBase III şi IV.

Comparativ cu standardul pieţii, versiunea beta a lui Foxpro 2.0 nu a strălucit la modul de comandă al imprimantelor. În meniul de imprimare există cîteva comenzi rudimentare pentru stabilirea marginilor, a orientării, etc., dar driver-ele speciale pentru imprimante matriciale şi laser lipsesc. Dacă se doreşte, de exemplu, listarea din Foxpro 2.0 pe o imprimantă HP-Laserjet, terminatorii de linie şi alte secvenţe de întreruperi trebuie realizate cu secvenţe de comenzi. Pentru profesionişti acest lucru nu constituie o problemă, dar începătorilor nu le este tocmai uşor.

După informaţiile de la Foxsoftware, pînă la lansarea pe piaţă, în această vară, vor mai fi aduse cîteva îmbunătăţiri la acest modul.

Generatorul de rapoarte, controlabil în întregime prin meniuri, poate lucra cu variabile, cu cîmpuri suplimentare de calcul (abatere, varianţă) şi poate repeta antetul de grup pe fiecare pagină. Salutară este şi opţiunea care permite restartarea numărului paginii la schimbarea grupului. Definiţia unui raport poate fi memorată.

Introducerea şi poziţionarea de cîmpuri şi de texte în interiorul unui raport se poate face comod cu ajutorul mouse-ului.

Pe lîngă tehnica ferestrelor cunoscută din versiunile anterioare, Foxpro dispune de o serie de utilitare care în viaţa de zi cu zi se pot dovedi a fi foarte folositoare.

Pentru operaţiile obişnuite cu fişiere utilizatorul poate utiliza managerul fişierelor care permite copierea, ştergerea, mutarea, fişierelor sau a unor întregi directoare.

Un calculator de buzunar ajută la efectuarea unor calcule intermediare, de asemenea mai fiind disponibile şi o tabèlă ASCII şi un calendar. Pentru destindere este oferit un joc de puzzle.

Foarte folositor se dovedeşte a fi şi clipboard-ul (suportul, magazia intermediară). Părţi ale conţinutului ecranului pot fi astfel "captate" şi inserate astfel în alte tabele. În acest mod se elimină necesitatea retastării unor date. Acelaşi lucru este valabil, într-o anumită măsură şi pentru descrierile de rapoarte. Părţile standard pot fi reutilizate cu ajutorul comenzilor "Cut and Paste".

După cum se ştie, Foxpro nu este doar un SGBD foarte uşor de învăţat şi foarte rapid ci şi un instrument confortabil pentru proiectanţii de aplicaţii cu baze de date. Pentru aceştia din urmă Foxpro 2.0 are implementate cîteva noutăţi interesante.

Printre acestea se numără structura deschisă a programelor, care permite programatorilor cuplarea prin intermediul API (Application Program Interface) unor rutine externe scrise în C sau Assembler.

În acest mod pot fi realizate şi funcţii pe care propriul limbaj de programare - un fel de dBase extins - nu le poate realiza. În plus pot fi utilizate acum şi biblioteci "externe" standardizate, ceea ce duce desigur la ridicarea productivităţii muncii. Avantajoasă este şi utilizarea generatoarelor de machete şi meniuri.

Generatorul de machete lucrează cu obiecte care pot fi înglobate în ecrane şi în meniuri. Sursa program care trebuie executată pentru prezentarea machetelor şi meniurilor este generată fără complicaţii. Suprafaţa utilizator Foxpro 2.0 poate fi astfel extinsă cu uşurinţă cu puncte suplimentare în meniuri şi cu ecrane suplimentare şi poate fi combinată cu alte programe şi rapoarte.

Proiectarea este supravegheată de managerul de proiecte care permite documentarea tuturor programelor, măştilor, meniurilor, etc., care sînt necesare pentru generarea unei aplicaţii. Fişierul proiect poate fi utilizat pentru crearea unei aplicaţii (.FXP) sau a unui program executabil (.EXE).

Foxpro 2.0 este un produs reuşit, noua versiune fiind recomandabilă nu numai ca o îmbunătăţire pentru fanii de Foxpro ci şi ca o alternativă pentru dBase.

(R.M.)

Tabel comparativ

Program	Foxpro 2.0	dBase IV/1.1	Paradox 3.5
Producător	Foxsoftware	Ashton-Tate	Borland
Preţ (DM)	Un post de lucru 2502, LAN (pt. fiecare server) 4098	2679	2451
Meniuri pull-down	da	da	nu
Număr articole	1 miliard	1 miliard	2 miliarde
Dimensiune maximă articol	4000 Byte	4000 Byte	4000 Byte
Număr cîmpuri	255	255	255
Limbaj de programare	da	da	da
Suport SQL	da	da	cu SQL-Link
Protecţie prin parolă	da	da	da
Integrare de imagini	da	nu	nu
Suport Windows 3.0	nu	nu	nu
Manager fişiere	da	nu	nu
Tehnica ferestrelor	da	nu	nu
Tehnici speciale de indexare	da	da	nu

Recunoaşterea scrisului

Reţele neuronale

Partea a IV-a

În ultimii ani, recunoaşterea scrisului prin calculator s-a perfecţionat mereu. Recunoaşterea de caractere prin intermediul reţelelor neuronale este un pas într-o direcţie promiţătoare.

Cuprinsul cursului:

- Partea 1: Privire de ansamblu asupra temei
- Partea 2: Teoria algoritmului propagării înapoi
- Partea 3: Sisteme expert şi asociativitate
- **Partea 4: Recunoaşterea scrisului**

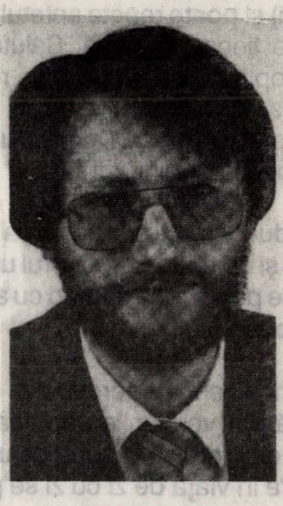
La prima vedere pare ciudat ca recunoaşterea de caractere să reprezinte pentru calculator o problemă foarte dificilă, cînd omul nu are nici un fel de dificultăţi şi doar şi calculatorul lucrează numai cu cifre şi litere. Dacă acestea sînt introduse prin intermediul tastaturii, aceasta desigur nu este o problemă pentru maşină, deoarece recunoaşterea unei litere este făcută de om, iar prin tastare caracterul este transformat într-un cod care poate fi prelucrat de calculator (cod ASCII).

Dacă însă procesul de recunoaştere care are loc în timpul citirii trebuie executat de maşină, atunci în primul rînd trebuie "transformat" în informaţie numerică suportul iniţial - această pagină de revistă de exemplu. Acest lucru se face prin intermediul unei camere video sau cu un scanner, care parcurg rînd cu rînd suportul de pornire, preluînd informaţii cu ajutorul unor fotodiode. La echipamente ce lucrează alb-negru, diferenţele luminos - întunecat sînt transformate pentru reprezentare pe ecranul calculatorului în succesiuni de biţi: un bit setat corespunde unui punct întunecos, un bit nesetat unuia luminos. Deoarece aceleaşi litere, ca succesiuni de biţi, apar diferit funcţie de formă şi mărime, numai un algoritm performant este capabil să realizeze interpretarea.

Procesul de recunoaştere la om se bazează pe următoarele etape: prin acomodarea diafragmei, literele apar într-un soi de "formă normalizată", într-o diferenţiere cît mai netă faţă de fundal. Apoi, omului îi reuşeşte prin asociere să completeze litere incomplete şi să citească şi versiuni deformate. Această capacitate este aplicată şi la cuvinte întregi. Deoarece omul foloseşte întotdeauna şi contextul cuvintelor în propoziţii sau al literelor în cuvinte, caractere asemănătoare, cum ar fi "1" şi "l", se deosebesc prin faptul că cifra "1" nu apare niciodată în interiorul unui cuvînt în texte obişnuite, iar litera "l" nu apare izolat sau în cadrul unui număr.

Recunoaşterea corectă a caracterelor

Sarcina programelor OCR (Optical Character Recognition) este de a simula cît mai bine aceste caracteristici umane. În multe programe îşi găsesc aplicaţie mai ales două metode: suprapunerea de modele (pattern matching) şi identificarea de caracteristici (feature recognition). La suprapunerea de modele (compararea de forme - pattern matching) succesiunile de biţi generate de scanner sînt comparate, pixel cu pixul, cu litere "ideale", memorate în calculator. Deoarece caracterele scanate sînt însă foarte diferite în mărime sau lăţime, ele



Peter Schmitz a studiat matematica şi biologia; în paralel, s-a ocupat cu probleme de evoluţie relative la autoorganizarea proceselor vii. Lucrarea sa de diplomă (1988) se plasează undeva între geometria algebrică şi teoria numerelor. A realizat un soft de simulare al modului de lucru al reţelelor neuronale şi a fondat în 1989 o întreprindere care se ocupă de aplicarea metodelor inteligenţei artificiale în aplicaţii industriale.

trebuiesc mai întîi "normalizate" la dimensiuni unitare. Chiar şi în acest caz, imagini binare ale unui aceluiaşi caracter pot fi foarte diferite (vezi fig. 1). Soluţionarea acestei probleme se încearcă prin memorarea mai multor modele pentru fiecare literă şi admiterea unui domeniu de toleranţă. O toleranţă prea mare duce însă la identificări eronate. Dacă se memorează prea multe modele, devine necesară multă memorie şi creşte timpul de calcul.

În cadrul procedeului de identificare a caracteristicilor, literele sînt caracterizate prin proprietăţi topologice. De exemplu "C" este determinat printr-o linie, arcuită convex spre stînga. Apar însă şi sensuri multiple: atît cifra "0" cît şi litera "O"

sînt caracterizate printr-un drum închis. Înainte de a putea identifica literele prin caracteristici, secvenţele de biţi trebuiesc "skeletonizate". (Adică, de exemplu, dintr-o linie de 3 pixeli se obţine una de un singur pixel).

În ciuda anunţurilor de reuşită a diverşi producători, suporturile copiate le creează dificultăţi majore mai tuturor programelor OCR. La digitalizare sau sînt "lipite" prea multe litere, astfel încît nu sînt interpretate ca litere singulare, sau sînt atît de "sparte", încît dintr-un caracter apar mai multe. Chiar şi litere izolate corect sînt deseori percepute atît de deformat încît devin ilizibile. În timp ce segmentarea literelor, deci fragmentarea unui cuvînt în componentele sale individuale, depinde în primul rînd de performanţele altor procedee, reţelele neuronale îşi dovedesc eficienţa mai ales la identificarea caracterelor deformate sau "sparte".

Împreună cu programul de simulare NBP se livrează un exemplu de recunoaştere a scrisului cu care se pot face experimentări. Intrarea reţelei constă din 18x18 pixeli, iar ieşirea din 99 de celule, pentru recunoaşterea a 99 de caractere distincte. În exemplu nu se tratează însă "decuparea" unor caractere dintr-un text întreg, ci doar identificarea de mostre de pixeli, pentru a clasifica corect literele.

Fişiere de pixeli scannate în format IMG pot fi reprezentate pe ecran cu ajutorul unui program specific; acesta decupează litere dintr-un text şi le memorează într-un fişier propriu. Este posibilă şi micşorarea imaginii, pentru a reduce volumul de date. Funcţie de rezoluţia scannerului, pentru un caracter se obţin mai mulţi sau mai puţini pixeli. Caracterele memorate se folosesc ca mostră de antrenament pentru o reţea neuronală. Cum s-a mai spus, fiecărei mostre i se ataşează un model de ieşire (respectiv, apartenenţa la o celulă de ieşire). Deci de exemplu mostra de pixeli pentru "A" activează celula de ieşire 1, mostra "B" celula de ieşire 2 ş.a.m.d.

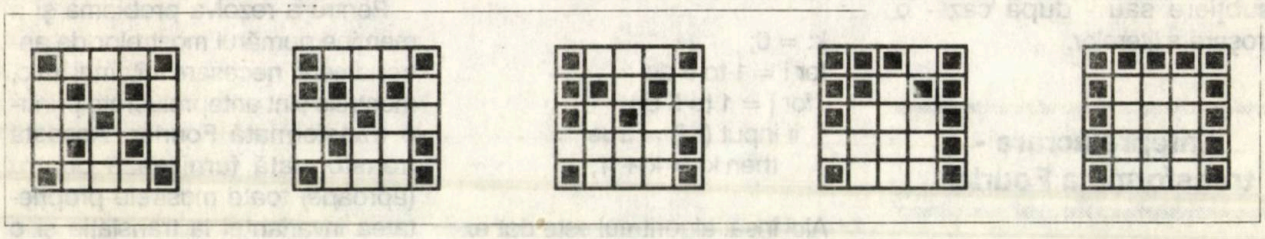


Figura 1 arată dificultăţile interpretării mostrelor de pixeli. Secvenţa reprezintă transformarea unui X într-un U. Dacă al doilea model din stînga încă poate fi recunoscut ca "X" şi al doilea model din dreapta încă poate fi interpretat ca "U", la modelul din centru sînt posibile noi interpretări (de exemplu, K sau W).

Aceste informaţii relative la scopul învăţării sînt memorate odată cu mostrele de pixeli. Pot fi imaginate desiguri şi alte codări ca funcţii obiectiv ale reţelei neuronale - de exemplu, direct valoarea ASCII corespunzătoare diverselor caractere. În acest caz ar fi necesare numai 3 celule de ieşire pentru reprezentarea zecimală, respectiv 8 pentru reprezentarea binară. Cercetări proprii au relevat însă faptul că acest mod de reprezentare a informaţiei constituie o îngreunare serioasă a procesului de convergenţă în faza de învăţare, mai ales datorită faptului că prin folosirea funcţiei sigmoide (vezi partea a II a acestei serii) stabilităţi se formează numai în jurul valorilor 0 şi 1. În plus, litere total distincte au coduri binare ASCII asemănătoare, ceea ce se răsfrînge negativ asupra asociativităţii necesare.

Sarcina fazei de învăţare este de a modifica ponderile din reţea în aşa fel încît să apară o configuraţie care satisface caracteristicile cerute ale fişierului de antrenament. Ceea ce deosebeşte reţelele neuronale de alte tehnici este faptul că nu mai este necesară o descriere explicită a algoritmului, astfel că această configuraţie se produce independent, în cursul fazei de învăţare. Proprietăţile ei sînt dirijate prin intermediul mostrelor de antrenament şi al funcţiilor obiectiv asociate.

Avînd odată o reţea "antrenată", datorită asociativităţii reţelelor neuronale, imagini binare asemănătoare ale unor litere sînt clasificate corect. Ceea ce, în cadrul codării menţionate iniţial, înseamnă că se activează o singură celulă de ieşire şi se dezactivează toate celelalte. Prin "mărimea" activităţii există o măsură suplimentară a apartenenţei la o clasă. Ea poate fi utilizată ca definiţie a domeniului de toleranţă: o mostră aparţine unei clase dacă valoarea de ieşire corespunzătoare este peste un anumit prag, iar toate valorile celelalte se găsesc sub un anumit prag. Dificil este însă a-ţi face o imagine clară despre noţiunea pomenită mai sus de "asemănare". Cît de mult trebuie să fie corelate două imagini binare, pentru a fi considerate "asemănătoare"?

Se pot cu uşurinţă imagina realizări foarte diferite ale unei aceleaşi litere, care diferă însă foarte mult prin amplasarea pixelilor. Pentru a nu fi nevoiţi, ca la "pattern matching", de a prelua toate combinaţiile posibile în fişierul de antrenament, se recomandă şi aici tehnici de anteprelucrare corespunzătoare. În programul de simulare NBP sînt realizate 3 tipuri.

În primul rînd, există posibilitatea translatării: imaginile de pixeli sînt duse în colţul din stînga-sus, iar coloanele şi rîndurile vide sînt pur şi simplu eliminate.

În al doilea rînd, în imaginile micşorate - şi micşorarea este de fapt o anteprelucrare - se poate varia pragul pentru amplasarea unui nou pixel. Astfel se realizează o subţiere sau - după caz - o îngroşare a literelor.

Anteprelucrare - transformata Fourier

Ca o a treia tehnică este oferită o transformată Fourier normată bidimensională. La intrări pătratice, ea are proprietatea deosebită a invarianţei la translaţie. Ceea ce înseamnă că indiferent unde se găseşte o imagine binară în cadrul unui cîmp de intrare, ea este proiectată totdeauna la fel.

Totuşi, chiar şi transformata Fourier rapidă (FFT) necesită calcule laborioase. (Ea se foloseşte totuşi, în mod standard, în electrotehnică datorită propietăţilor deosebite în analiza funcţiilor periodice - e drept, în acest domeniu, transformata Fourier unidimensională).

Ca suport pentru recunoaşterea scrisului, datorită consumului de timp, ea se recomandă numai cu un hardware specializat. Se poate pune întrebarea la ce mai e pomenită atunci, în acest context. Deoarece în această serie de articole nu poate fi vorba de prezentarea unor soluţii complete pentru probleme speciale, ci doar de introducerea în tehnici noi de prelucrare a informaţiilor, cititorului i se oferă un prim contact care să-i permită manipularea corectă a acestora, cîteva puncte de plecare posibile pornind de la exemple.

Cazul combinaţiei transformată Fourier/propagare înapoi este discutat pe exemplul aflării numărului de pixeli setaţi într-o suprafaţă predeterminată. Fie dată o suprafaţă de 5x5 pixeli. În această suprafaţă de 25 pixeli se admit pînă la 5 pixeli "setaţi" - avînd valoare logică "true". Reţelei neuronale i se cere să "numere" corect pixelii setaţi.

Este evident că pe alte căi soluţia se obţine mult mai uşor şi mai repede. O buclă simplă de contorizare ar putea fi:

```
k: = 0;
for i = 1 to 5 do
  for j = 1 to 5 do
    if input (i, j) = true
      then k: = k + 1;
```

Aici însă, algoritmul este dat explicit. Spre deosebire de aceasta, reţeaua trebuie să "găsească" singură algoritmul, pornind de la cîteva (puţine) mostre de antrenament!

Cu NBP se produce o reţea de 5X5 celule de intrare şi 5 celule de ieşire, pentru clasele "un pixel", "doi pixeli", pînă la "5 pixeli".

Problema din ultimul articol, de a minimiza numărul mostrelor de antrenament, este reluată odată cu acest exemplu.

Mai întîi, cîteva consideraţii preliminare relativ la posibilităţile de combinare: pentru a plasa un pixel într-o suprafaţă de 25 de cîmpuri diferite, există 25 de posibilităţi. Pentru 2 pixeli, există 25x24/2 = 300 de posibilităţi, deoarece nu importă ordinea plasării. Corespunzător, pentru 3 pixeli există 25x24x23/3/2 = 2300, pentru 4 pixeli 12.650 şi pentru 5 pixeli chiar 53.130 de combinaţii.

Pentru a rezolva problema şi a menţine numărul mostrelor de antrenament necesare cît mai mic, mostrele sînt anteprelucrate printr-o transformată Fourier. Această transformată furnizează pentru (aproape) toate mostrele proprietatea invarianţei la translaţie şi o invarianţă la oglindire axială pe axa x. După transformare Fourier, mostrele din figura 3 sînt identice.

Spre deosebire de translaţia pomenită mai sus, la care numai pixelii nesetaţi din stînga şi de sus sînt eliminaţi, valorile cîmpurilor nu mai constau în continuare din valori de "0" şi de "1", respectiv stările "adevărat şi fals", ci din transformate Fourier, deci numere în virgulă mobilă. Avantajul metodei este determinat de toleranţa mai mare la erori - cîte un pixel rătăcit, nedorit, nu are efecte aşa de mari ca şi la translaţie. Dezavantajos este calculul laborios ce trebuie efectuat.

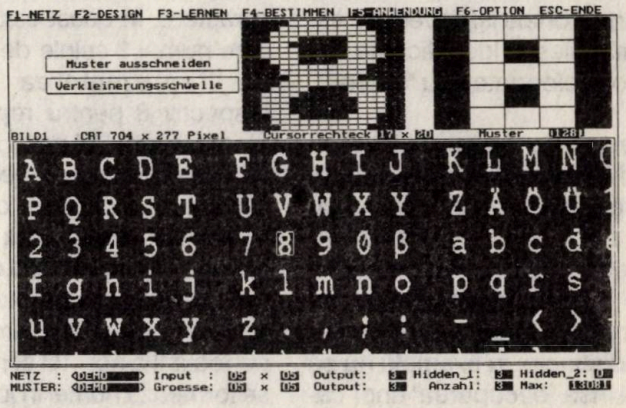


Figura 2 arată ecranul vizibil din NBP la prelucrarea de imagini scannate. Cu dreptunghiul cursor reprezentat mărit în partea superioară, se poate decupa o mostră. Aceasta este apoi micşorată la dimensiunea cerută la intrare, folosind un prag selectabil. Rezultatul micşorării este afişat pe ecran în colţul din dreapta sus.

Probleme cu posibilităţi foarte mari de combinare se abordează cel mai bine în felul următor: după selecţia unei anteprelucrări potrivite, din fiecare clasă (2,3,4,5 pixeli) se memorează cîteva mostre şi se antrenează reţeaua cu acestea. Din pixelii singulari nu există mostre diferite, deoarece transformata Fourier este totdeauna aceeaşi. reţeaua se comportă aşa cum pare "de bun simţ": activează celula care se apropie cel mai mult de mostra dată, şi anume celula 5.

Fără combinarea cu transformata Fourier, efortul de calcul în faza de învăţare ar fi disproporţionat de mare, deoarece ar fi fost de învăţat şi verificat posibilităţi de amplasare a pixelilor din cele mai diverse.

Sînt reţelele neuronale drumul spre "tehnica inteligentă"? Înainte de a răspunde la această întrebare, ar trebui clar definit ce se înţelege prin inteligenţă. Există oameni care susţin cu toată seriozitatea că o sită este inteligentă: deoarece ştie să deosebească ceaiul de frunzuliţele din el. Ca utilizator de unelte pentru rezolvarea unor probleme reale ale industriei este însă mai interesantă întrebarea - cu ce este mai performantă această tehnică decît altele, uzuale?

Răspunsul la aceasta depăşeşte cadrul acestei serii, în care pe baza unor noţiuni de bază sumare au fost prezentate cîteva exemple de aplicare. Autorul are experienţa unor produse finalizate în aceste domenii şi a realizat cîteva aplicaţii; cei interesaţi pot obţine desigur informaţii suplimentare direct de la el.

Peter Schmitz

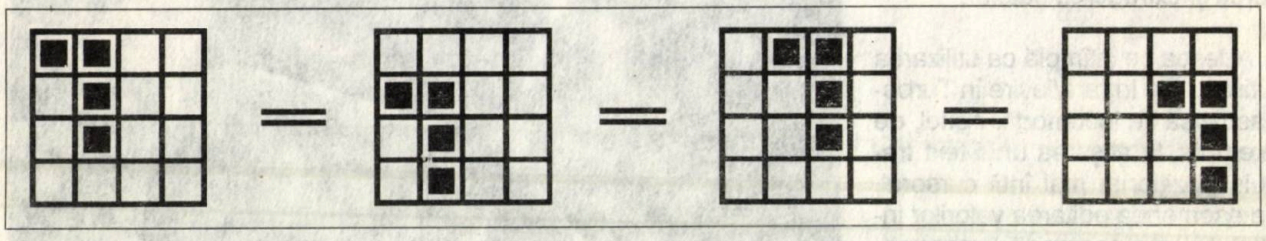


Figura 3. Transformata Fourier proiectează identic modele translatate. Rezultatul transformării se foloseşte ca intrare într-o reţea neuronală. În acest fel se obţine un fel de preclasificare.

După terminarea cu succes a învăţării, se testează comportarea reţelei cu mostre care îi sînt necunoscute. Dacă acestea sînt recunoscute corect, atunci ela au fost deja asimilate de reţea: reţeaua "generalizează" deci datele de antrenament cunoscute. Dacă noile mostre sînt identificate eronat, ele se adaugă la fişierul de antrenament, reluîndu-se faza de "învăţare" a reţelei. Continuînd în felul acesta, se pot acoperi toate combinaţiile posibile ale problemei. În exemplul de mai sus, sînt necesare doar 22 (!) de mostre de antrenament pentru a rezolva corect toate cele 25 + 300 + 2300 + 12650 + 53130 = 68.405 combinaţii posibile.

Reducerea timpului de calcul

După cum arată deja exemplul de mai sus, printr-o combinare adecvată cu alte tehnici, (spre deosebire de exemplul din ultimul articol), se pot obţine reduceri substanţiale de date în fişierul de antrenament şi respectiv timpi de învăţare mult reduşi.

În afară de aceasta, în situaţii neprevăzute - ca de exemplu 6 sau 7 pixeli în cîmpul de intrare -

micro ATCI S.R.L.

Micro ATCI S.R.L.
4300 Tîrgu Mureş
C.P. 64, O.P.1
tel. 954/31660

oferă

celor interesaţi în **rezolvarea**, pe calculatoarele personale compatibile IBM-PC, a

sistemelor mari de ecuaţii lineare

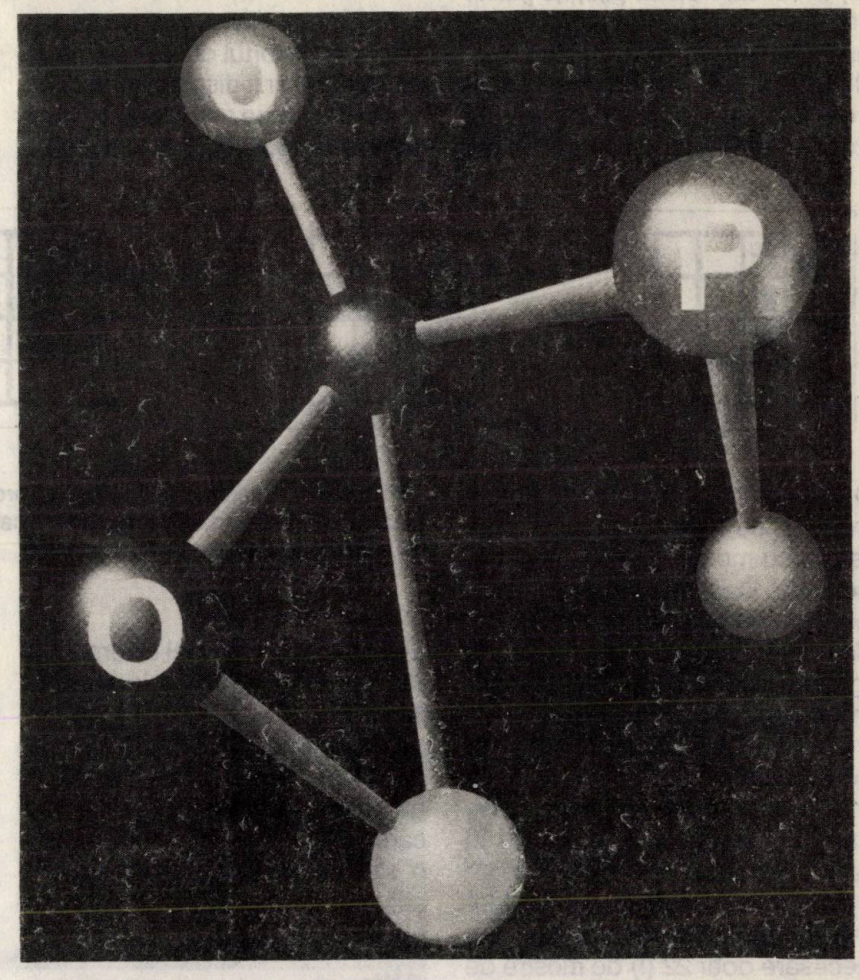
SISREZ v1.0

un pachet de programe cu performanţe ridicate privind viteza şi precizia de execuţie.

Încă o dată de la început ?

OOP - Partea a patra

Cu BS_Tools va fi prezentată o mică ierarhie de obiecte care îmbunătăţeşte confortul la introducerea şi extragerea datelor.



Adesea se întîmplă ca utilizarea rutinelor de intrare/ieşire în Turbo-Pascal să fie incomodă. Astfel, de exemplu, la afişarea unui text trebuie poziţionat mai întîi cursorul, de asemenea editarea valorilor introduse este incomodă. Programatorii se văd mereu puşi în faţa situaţiei de a "scoate" mai mult din rutinele de bază. Ar fi foarte bine deci dacă la nevoie ar putea fi accedat un Unit care să ofere confortul necesar. Ar fi de imaginat rutine care să poată scrie şi citi de pe ecran şiruri de caractere sau de cifre. În acelaşi timp se doreşte ca problema extensiilor să rămînă deschisă. Ulterior urmează ca modul de prezentare al aplicaţiei să fie îmbunătăţit prin introducerea ferestrelor şi a boxelor de dialog.

Dar cine nu cunoaşte deja problema pe care o implică astfel de extensii? Se modifică ceva într-un anumit loc şi dintr-o dată nu mai funcţionează nici un program care utiliza acel Unit.

Cu ajutorul obiectelor astfel de probleme pot fi evitate. Singura condiţie este ca obiectele deja existente să nu fie modificate funcţional. Modificările trebuie să se axeze deci pe "ce" face un obiect. "Cum" face o metodă ceva, trebuie păstrat. Ceea ce este funcţional nou se defineşte oricum în noile tipuri de obiecte. Atunci cînd este necesar, aceste însuşiri pot fi preluate de la obiectele disponibile.

Dacă se doreşte programarea orientată obiect a tuturor uneltelor care îmbunătăţesc confortul utilizării intrărilor şi ieşirilor, atunci trebuie găsite similitudinile şi diferenţele. Caracteristicile identice pot fi definite în obiectul mamă, tipurile speciale de obiecte moştenind aceste caracteristici. Similitudinile nu trebuie căutate îndelung. Toate obiectele necesită metode care să le permită prezentarea pe ecran şi îndepărtarea lor de acolo.

Dar este evident că un şir de caractere va necesita o altă rutină pentru reprezentare decît o fereastră. Comun este faptul că toate obiectele au nevoie de o metodă pentru a putea fi prezentate. Modul în care vor trebui implementate aceste metode va fi diferit.

Unit-ul care va prelua aceste tipuri de obiecte se va denumi **BS_Tools**. O primă prezentare a lui este prezentă în "Listing 1" şi reprezintă fundamentul pentru o ierarhie de obiecte.

Comun tuturor obiectelor din Unit este faptul că ele trebuie să poată fi prezentate în poziţia dorită de pe ecran. Obiectul mamă în acest caz ar putea fi **BS_Loc**. El defineşte două tipuri de date **X** şi **Y**. Ele sînt identice cu poziţia ulterioară de intrare sau de ieşire. Metoda **Init** este utilizată pentru stabilirea poziţiei. Cu **GetX** şi **GetY** poate fi interogată poziţia momentană a unui obiect pe ecran. Pot fi imaginate şi alte funcţii - cum ar fi de exemplu poziţionarea cursorului. Din motive de spaţiu ne vom limita însă la funcţiile anterior enumerate.

S-ar putea crede că metoda **Init** este inutilă, ea nu face altceva decît să transmită date, în rest nu se întîmplă nimic. Şi totuşi nu este aşa. Definiţia tipului **Obiect** (partea

întîi a acestui curs) exprimă clar faptul că un obiect este o reuniune a datelor şi a metodelor de prelucrare *permise* asupra lor. "Dezavantajul" faptului că datele trebuie transmise mai întîi obiectului este compensat de faptul că datele interne nu pot fi influenţate decît de metodele cărora li se permite acest lucru. Acest fenomen - denumit încapsulare a datelor - este tipic pentru programarea orientată obiect. Încapsularea poate fi întărită dacă definiţiile obiectelor sînt date în partea Interface, iar partea de cod a metodelor aparţinătoare în partea Implementation. Un utilizator care are la dispoziţie doar partea "publică" a Unit-ului, trebuie să-şi transmită datele spre obiect prin metodele Init disponibile. Altă soluţie nu există. În acest mod datele interne sînt protejate împotriva unor modificări nepermise.

O a doua caracteristică foarte importantă a obiectelor este posibilitatea moştenirii caracteristicilor. Se defineşte deci un nou tip de obiect, care "se trage" dintr-altul. Cu aceasta noul obiect preia toate însuşirile deja definite în obiectul mamă. Tipul de obiect **BS_Semn** este derivat din **BS_Loc**, vizibil prin numele obiectului mamă scris între paranteze. Un obiect de tipul **BS_Semn** poate deci dispune de datele şi metodele obiectului **BS_Loc**.

Economie de text sursă

Deja în partea Implementation din Unit este utilizat acest lucru. Metoda **Init** a obiectului **BS_Semn** utilizează metoda Init a obiectului mamă.

Şi metoda **Move**, pe lîngă două metode proprii, foloseşte şi metoda Init a lui **BS_Loc**.

Cu ajutorul unui mic program de test se poate stabili imediat că şi sarcinile lui **GetX** şi **GetY** sînt înţelese de un obiect de tipul **BS_Semn**. Pe lîngă efectul protejării datelor împotriva unor accese nedorite prin încapsulare, se economiseşte şi cod, deoarece metodele comune nu trebuie redefinite.

Acest prim exemplu (Listing 1) care se sprijină puternic pe primele părţi ale acestui curs este însă foarte subţire încă. Cu excepţia scrierii, ştergerii şi mutării unui caracter altceva nu se poate face.

Din acest motiv Unit-ul **BS_Tools** este extins în "Listing 2". Nou este aici obiectul **BS_String**. Metodele **Init**, **Draw** şi **Delete** sînt definite din nou sau altfel exprimate: ele au fost adaptate pentru lucrul cu şiruri. Cu **Get_Str** poate fi stabilită valoarea parametrului intern **Str**. În maniera obişnuită, au fost utilizate metodele obiectului mamă pentru a se economisi cod. Metoda **Move** n-a fost declarată încă o dată pentru **BS_String** deoarece codul ei a rămas acelaşi. Trebuie acceptat deci faptul că metodele obiectului mamă pot fi preluate pe nevăzute.

Cu un mic program de test se poate observa că rutinele **BS_String.Draw** şi **BS_String.Delete** îşi realizează în totalitate funcţiile. Numai mutarea textelor nu se face tocmai corect. Pe noua poziţie de pe ecran va apare numai primul caracter din şir. Ce este greşit?

Sursa program propriu-zisă este corectă. O instanţiere a lui **BS_String** apelează rutina **Move**, în care prima instrucţiune este **Delete**. Dar despre care metodă de ştergere este vorba? Instanţierea ar dori să lanseze în execuţie metoda **BS_String.Delete**. În loc de aceasta se execută însă cea din **BS_Semn**. Acelaşi lucru se întîmplă şi în următoarea instrucţiune a programului. Din nou este apelată o metodă care nu se potriveşte instanţierii. Acest lucru poate fi explicat astfel: **Move** este o metodă a lui **BS_Semn**, deci compilatorul stabileşte faptul că **Draw** şi **Delete** sînt metode care aparţin lui **BS_Semn**. Urmarea este că numai un caracter al şirului va fi mutat.

Trebuie deci programată o metodă proprie pentru **BS_String**, cu toate că nu se modifică nimic în textul sursă? Normal că nu trebuie. Cu un mecanism pentru metode virtuale această problemă poate fi rezolvată. Acest mecanism nu va face altceva decît va lăsa liber în momentul compilării care metodă **Delete** şi care metodă **Draw** urmează a fi utilizată. Aceasta va fi stabilit doar atunci cînd tipul instanţierii apelante este cunoscut.

O metodă pentru toţi

Acest mecanism nu funcţionează însă de la sine. El trebuie iniţializat. Dacă se uită acest lucru, în locul saltului la metoda dorită se va afla altceva. Urmarea poate fi o cădere a sistemului. Compilatorul nu poate stabili însă dacă o instanţiere cu metode virtuale a fost iniţializată. Răspunderea, ca mecanismul metodelor virtuale să fie iniţializat pentru fiecare instanţiere apelantă, cade pe umerii utilizatorului.

Iniţializarea are loc printr-un apel special de procedură. În antetul de procedură cuvîntul **Procedure** este înlocuit cu cuvîntul **Constructor**. Este avantajos să se modifice funcţionarea unei proceduri deja existente cu constructor, care să fie apelată în orice caz înaintea unei metode virtuale. În Unit-ul **BS_Tools** acestea sînt rutinele **Init** ale obiectelor **BS_Semn** şi **BS_String**. În ambele obiecte, metodele **Draw** şi **Delete** trebuie să fie virtuale. Este suficientă adăugarea cuvîntului **Virtual** în declaraţie.

Dacă se compară "Listingul 3" cu "Listingul 2" se observă imediat unde au fost operate modificări. Un mic program de test va funcţiona mulţumitor în acest caz. Unit-ul **BS_Tools** din "Listingul 2" poate fi abandonat.

Fără să fi scos în evidenţă pînă acum, a fost utilizată o altă caracteristică importantă a programării orientate obiect: ambele obiecte capabile de reprezentare de pînă acum folosesc identificări identice pentru a rezolva lucruri diferite.

Metoda **Draw** a obiectului **BS_Semn** face cu totul altceva decît metoda cu acelaşi nume a obiectului fiu **BS_String**. Cu **Delete** se întîmplă acelaşi lucru.

Această posibilitate, de a denumi acţiuni diferite cu acelaşi nume, este denumită polimorfism.

Pentru utilizarea metodelor virtuale mai există însă o restricţie. Nu numai numele parametrilor trebuie să corespundă ci şi tipul şi numărul acestora. Deoarece **Draw** şi **Delete** nu transmit parametri, în acest caz nu trebuie să ţinem cont de această restricţie.

Virtual şi dinamic

În partea a treia a acestui curs ne-am referit la metodele virtuale. Dorim să subliniem încă o dată că metodele virtuale cuplate cu programarea dinamică oferă posibilitatea extinderii programării orientate obiect.

În "Listing 4" este prezentată în acest scop ultima versiune a Unitului **BS_Tools**. În partea Interface sînt prezenţi, în faţa definiţiilor propriu-zise ale obiectelor, pointerii necesari.

Un obiect dinamic are avantajul că este creat doar atunci cînd este necesar. Mai ales atunci cînd sînt deschise simultan mai multe instanţieri ale unui tip apar avantajele gestionării dinamice a memoriei. La pornirea programului înafara celor 4 byte pentru fiecare pointer nu se rezervă altă memorie. Programatorul este cel care trebuie să gestioneze heap-ul (memoria disponibilă).

Dacă sînt necesare instanţieri atunci ele pot fi create şi eliminate din memorie cu ajutorul procedurilor deja cunoscute **New** şi **Dispose**. Metoda **Done**, inserată, serveşte la curăţarea corectă a heap-ului. Metoda **Done** este declarată cu un nou cuvînt cheie: **Destructor**. Sarcina propriu-zisă de eliberare a memoriei, în cazul lui Destructor, se execută în fundal. Chiar dacă metoda nu conţine nici o instrucţiune, ea îşi îndeplineşte menirea. Un Destructor poate fi imaginat ca un Unit, care nu exportă nimic, dar care prelucrează o parte de iniţializare.

Simultan în **BS_Tools** a fost definit un nou obiect: **BS_Intro**. El exportă trei noi date (I_String, I_Attr şi I_Length). Nucleul îl constituie funcţia **ReadString**. Cu ajutorul parametrilor transmişi prin intermediul constructorului **Init** este posibilă afişarea unui prompter într-o anumită poziţie pe ecran şi aşteptarea introducerii unui şir de caractere în dreapta acestuia. **NrMax** stabileşte lungimea maximă a şirului de editat. Este clară, şi în acest caz, utilizarea diferitelor metode moştenite.

Totuşi din motive de spaţiu vă prezentăm doar trunchiul unei rutine de introducere confortabile. Oricine poate reproduce ceea ce se întîmplă în aceste rutine, n-ar trebui să aibă probleme în a-i face adăugirile necesare.

Obiectul **BS_Intr** poate constitui punctul de plecare pentru o multitudine de obiecte speciale. Sînt de imaginat astfel obiecte care să permită introducerea şi editarea de numere de tipul Real sau LongInt. Obiectul ar putea prelua intern convertirea în formatul dorit. Imaginabilă este şi stabilirea unui domeniu pentru valorile de intrare. Astfel ar putea fi introduse numere şi s-ar putea simultan valida încadrarea lor în domeniul impus.

Obiectele mamă nu trebuie modificate

BS_Intr nu este însă un obiect tocmai curat programat. Cîteva metode moştenite nu funcţionează corect. Nefiind utilizabile trebuie definite din nou. Dar şi metoda **ReadString** trebuie modificată în cîteva locuri pentru ca datele interne să fie corect poziţionate.

Pentru cititorii care doresc să încerce cîteva din modificările respective, cîteva sfaturi vor fi binevenite: pentru ca metoda **Move** să poată fi folosită în anumite comdiţii, parametrii interni X şi Y trebuie să rămînă nemodificaţi după apelul lui **ReadString**. Deoarece obiectul nu poate fi mutat în timpul editării este suficient dacă şirul prompter şi şirul de intrare sînt aranjate pentru mutare. Ele trebuie colectate deci împreună în parametrul intern Str.

Faptul că **Move** nu funcţionează corect din cauza lui **ReadString** se poate observa cu ajutorul scurtului exemplu din "Listing 5". În modul "Trace Into" al debugger-ului se pot depista şi corecta uşor anomaliile: deoarece **Init** a fost declarat constructor, în metoda **Move** a lui BS_Semn (moştenire după două generaţii!) sînt utilizate deja metodele corecte **Draw** şi **Delete**. Problema ţine deci de utilizarea corectă a datelor interne.

Datorită exemplului din "Listing 5" a devenit evident cît de mult a crescut confortul cu această rutină. Cu puţine linii program se poate introduce un şir de caractere în orice poziţie dorită de pe ecran. Dacă trebuie introduse mai multe date pot fi deschise simultan mai multe obiecte. Atît timp cît acestea există în heap, şirul de intrare trebuie să fie oricînd disponibil. El poate fi afişat deci încă o dată la nevoie. De remarcat sînt şi procedurile extinse New şi Dispose. Primul parametru din paranteză specifică instanţierea pentru care trebuie rezervat sau eliberat spaţiu. Al doilea parametru specifică rutina cu care va fi executat procesul.

Obiectele odată definitivate mai aduc un avantaj programatorului. El nu mai trebuie să ştie decît ce fac obiectele. Cum funcţionează obiectele intern rămîne necunoscut utilizatorului şi nici nu-l interesează. Este suficient, de exemplu, să se ştie că obiectul **BS_String** afişează un şir în poziţia dorită de pe ecran. Dacă **BS_String** rezolvă acest lucru prin utilizarea procedurilor standard **Read** şi **Write** sau - aşa cum este cazul lui **BS_Tools** - prin scriere directă în memoria ecran, este neesenţial.

Este evident faptul că abia utilizarea complexă a tuturor tehnicilor programării orientate obiect aduce avantaje efective. Pentru un începător este indicată familiarizarea pas cu pas cu particularităţile OOP. Utilizarea debugger-ului integrat este cea mai edificatoare asupra proceselor interne.

(R.M.)

```
Unit Bs_Tools;
Interface

Const
 SegBWS = $B800; {Color}
{SegBWS = $B000;   Mono }
 OfsBWS : Word = $0000;

Type
   BS_Loc = Object
     X,Y : Integer;
     Procedure Init (xPos, yPos : Integer);
     Function GetX : Integer;
     Function GetY : Integer;
   End;
   BS_Semn = Object (BS_Loc)
     Semn : Char;
     Attr : Byte;
     Procedure Init (xPos, yPos : Integer;
       Ch : Char; Co : Byte);
     Procedure Draw;
     Procedure Delete;
     Procedure Move (xNou, yNou : Integer);
   End;

Implementation
Procedure BS_Loc.Init
            (xPos, yPos : Integer);
Begin
   X := xPos;
   Y := yPos;
End;
Function BS_Loc.GetX : Integer;
Begin
   GetX := X;
End;
Function BS_Loc.GetY : Integer;
Begin
   GetY := Y;
End;
Procedure BS_Semn.Init
              (xPos, yPos : Integer;
        Ch : Char; Co : Byte);
Begin
   BS_Loc.Init (xPos,yPos);
   Semn := Ch;
   Attr := Co;
End;

Procedure BS_Semn.Draw;
Begin
   OfsBWS := 2*(X-1)+160*(Y-1);
   Mem[SegBWS : OfsBWS+1] := Attr;
   Mem[SegBWS : OfsBWS]  := Ord(Semn);
End;

Procedure BS_Semn.Delete;
Begin
   OfsBWS := 2*(X-1)+160*(Y-1);
   Mem[SegBWS : OfsBWS] := 32;
End;

Procedure BS_Semn.Move
          (xNou, yNou : Integer);
Begin
   Delete;
   BS_Loc.Init (xNou,yNou);
   Draw;
End;

End.
```

{Listing 1. BS_Tools - o prima prezentare}

```
Unit Bs_Tools;

Interface

Const
 SegBWS = $B800; {Color}
{SegBWS = $B000;   Mono }
 OfsBWS : Word = $0000;

Type

   BS_Loc = Object
     X,Y : Integer;
     Procedure Init (xPos, yPos : Integer);
     Function GetX : Integer;
     Function GetY : Integer;
   End;

   BS_Semn = Object (BS_Loc)
     Semn : Char;
     Attr : Byte;
     Procedure Init (xPos, yPos : Integer;
       Ch : Char; Co : Byte);
     Procedure Draw;
     Procedure Delete;
     Procedure Move (xNou, yNou : Integer);
   End;

   BS_String = Object (BS_Semn)
     Str : String;
     Procedure Init (xPos, yPos : Integer;
       S : String; Co : Byte);
     Procedure Draw;
     Function Read_BS : String;
     Procedure Delete;
   End;

Implementation

Procedure BS_Loc.Init
            (xPos, yPos : Integer);
Begin
   X := xPos;
   Y := yPos;
End;

Function BS_Loc.GetX : Integer;
Begin
   GetX := X;
End;

Function BS_Loc.GetY : Integer;
Begin
   GetY := Y;
End;

Procedure BS_Semn.Init
              (xPos, yPos : Integer;
        Ch : Char; Co : Byte);
Begin
   BS_Loc.Init (xPos,yPos);
   Semn := Ch;
   Attr := Co;
End;

Procedure BS_Semn.Draw;
Begin
   OfsBWS := 2*(X-1)+160*(Y-1);
   Mem[SegBWS : OfsBWS+1] := Attr;
   Mem[SegBWS : OfsBWS]  := Ord(Semn);
End;

Procedure BS_Semn.Delete;
Begin
   OfsBWS := 2*(X-1)+160*(Y-1);
   Mem[SegBWS : OfsBWS] := 32;
End;

Procedure BS_Semn.Move
          (xNou, yNou : Integer);
Begin
   Delete;
   BS_Loc.Init (xNou,yNou);
   Draw;
End;

Procedure BS_String.Init
              (xPos, yPos : Integer;
       S : String; Co : Byte);
Begin
   BS_Loc.Init (xPos,yPos);
   Str := S;
   Attr := Co;
End;

Procedure BS_String.Draw;
Var i,s : Integer;
   Ch : Char;
Begin
   s := X;
   For i := 1 To Length (Str) Do Begin
     BS_Semn.Init ((i-1)+s,Y,Str[i],Attr);
     BS_Semn.Draw;
   End;
   X := s;
   Semn := Str[1];
End;

Function BS_String.Read_BS : String;
Begin
   Read_BS := Str;
End;

Procedure BS_String.Delete;
Const SfirsitSemn : Set Of Char = [Chr(0),
          Chr(9),Chr(10),Chr(12),Chr(13)];
Var i,j : Integer;
Begin
   i := X;
   j := Y;
   Repeat
     OfsBWS := 2*(i-1)+160*(j-1);
     If Chr (Mem[SegBWS : OfsBWS]) in
          SfirsitSemn Then Exit;
     BS_Semn.Delete;
     Inc(i);
   Until i = 80;
End;

End.
```

{Listing 2. BS_Tools -intr-o versiune }
{ imbunatatita, dar mutarea de}
{ siruri de caractere inca nu functioneaza}

```
Unit Bs_Tools;

Interface

Uses Crt;

Const
 SegBWS = $B800; {Color}
{SegBWS = $B000;  Mono }
 OfsBWS : Word = $0000;

Type

  BS_Loc = Object
    X,Y : Integer;
    Procedure Init (xPos, yPos : Integer);
    Function GetX : Integer;
    Function GetY : Integer;
  End;

  BS_Semn = Object (BS_Loc)
    Semn : Char;
    Attr : Byte;
    Constructor Init (xPos, yPos : Integer;
      Ch : Char);
    Procedure Draw; Virtual;
    Procedure Delete; Virtual;
    Procedure Move (xNou, yNou : Integer);
  End;

  BS_String = Object (BS_Semn)
    Str : String;
    Constructor Init (xPos, yPos : Integer;
      S : String);
    Procedure Draw; Virtual;
    Function GetStr : String;
    Procedure Delete; Virtual;
  End;

Implementation

Procedure BS_Loc.Init (xPos, yPos : Integer);
Begin
  X := xPos;
  Y := yPos;
End;

Function BS_Loc.GetX : Integer;
Begin
  GetX := X;
End;

Function BS_Loc.GetY : Integer;
Begin
  GetY := Y;
End;

Constructor BS_Semn.Init
            (xPos, yPos : Integer;
       Ch : Char);
Begin
  BS_Loc.Init (xPos,yPos);
  Semn := Ch;
End;

Procedure BS_Semn.Draw;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS+1] := TextAttr;
  Mem[SegBWS : OfsBWS]  := Ord(Semn);
End;

Procedure BS_Semn.Delete;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS] := 32;
End;

Procedure BS_Semn.Move
            (xNou, yNou : Integer);
Begin
  Delete;
  BS_Loc.Init (xNou,yNou);
  Draw;
End;

Constructor BS_String.Init
            (xPos, yPos : Integer;
       S : String);
Begin
  BS_Loc.Init (xPos,yPos);
  Str := S;
End;

Procedure BS_String.Draw;
Var i,s : Integer;
  Ch : Char;
Begin
  s := X;
  For i := 1 To Length (Str) Do Begin
    BS_Semn.Init ((i-1)+s,Y,Str[i]);
    BS_Semn.Draw;
  End;
  X := s;
  Semn := Str[1];
End;

Function BS_String.GetStr : String;
Begin
  GetStr := Str;
End;

Procedure BS_String.Delete;
Const Terminator : Set Of Char = [Chr(0),
      Chr(9),Chr(10),Chr(12),Chr(13)];
Var i : Integer;
Begin
  i := X;
  Repeat
    OfsBWS := 2*(X-1)+160*(Y-1);
    If Chr (Mem[SegBWS : OfsBWS]) in
                    Terminator
      Then Begin
        X := i;
        Exit;
      End;
    Bs_Loc.Init(X,Y);
    BS_Semn.Delete;
    Inc(X);
  Until X = 80;
  X := i;
End;

End.

{Listing 3. Scheletul de baza al unei }
{           ierarhii de obiecte }
```

```
Unit Bs_Tools;

Interface

Uses Crt;

Const
 SegBWS = $B800; {Color}
{SegBWS = $B000;  Mono }
 OfsBWS : Word = $0000;

Type

  BS_LocPtr  = ^BS_Loc;
  BS_SemnPtr  = ^BS_Semn;
  BS_StringPtr = ^BS_String;
  BS_IntrPtr  = ^BS_Intr;

  BS_Loc = Object
    X,Y : Integer;
    Procedure Init (xPos, yPos : Integer);
    Function GetX : Integer;
    Function GetY : Integer;
  End;

  BS_Semn = Object (BS_Loc)
    Semn : Char;
    Constructor Init (xPos, yPos : Integer;
      Ch : Char);
    Procedure Draw; Virtual;
    Procedure Delete; Virtual;
    Procedure Move (xNou, yNou : Integer);
    Destructor Done; Virtual;
  End;

  BS_String = Object (BS_Semn)
    Str : String;
    Constructor Init (xPos, yPos : Integer;
      S : String);
    Procedure Draw; Virtual;
    Function GetStr : String;
    Procedure Delete; Virtual;
  End;

  BS_Intr = Object (BS_String)
    I_String : String;
    I_Attr  : Byte;
    I_Length : Byte;
    Constructor Init (xPos,yPos: Integer;
        Prompt : String;
        NrMax,Attr : Byte);
    Function ReadString : String;
  End;

Implementation

Procedure BS_Loc.Init (xPos, yPos : Integer);
Begin
  X := xPos;
  Y := yPos;
End;

Function BS_Loc.GetX : Integer;
Begin
  GetX := X;
End;

Function BS_Loc.GetY : Integer;
Begin
  GetY := Y;
End;
```

```
Constructor BS_Semn.Init (xPos, yPos : Integer;
        Ch : Char);
Begin
  BS_Loc.Init (xPos,yPos);
  Semn := Ch;
End;
Procedure BS_Semn.Draw;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS+1] := TextAttr;
  Mem[SegBWS : OfsBWS]  := Ord(Semn);
End;
Procedure BS_Semn.Delete;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS] := 32;
End;
Procedure BS_Semn.Move (xNou, yNou : Integer);
Begin
  Delete;
  BS_Loc.Init (xNou,yNou);
  Draw;
End;
Destructor BS_Semn.Done;
Begin
End;
Constructor BS_String.Init (xPos, yPos : Integer;
        S : String);
Begin
  BS_Loc.Init (xPos,yPos);
  Str := S;
End;

Procedure BS_String.Draw;
Var i,s : Integer;
  Ch : Char;
Begin
  s := X;
  For i := 1 To Length (Str) Do Begin
    BS_Semn.Init ((i-1)+s,Y,Str[i]);
    BS_Semn.Draw;
  End;
  X := s;
  Semn := Str[1];
End;
Function BS_String.GetStr : String;
Begin
  GetStr := Str;
End;
Procedure BS_String.Delete;
Const
    Terminator : Set Of Char = [Chr(0),Chr(9),
                    Chr(10),Chr(12),Chr(13)];
Var i : Integer;
Begin
  i := X;
  Repeat
    OfsBWS := 2*(X-1)+160*(Y-1);
    If Chr (Mem[SegBWS : OfsBWS]) in Terminator
      Then Begin
        X := i;
        Exit;
    End;
    Bs_Loc.Init(X,Y);
    BS_Semn.Delete;
    Inc(X);
  Until X = 80;
  X := i;
End;
Constructor BS_Intr.Init (xPos, yPos : Integer;
        Prompt : String;
        NrMax, Attr : Byte);
Begin
  BS_String.Init (xPos,yPos,Prompt);
  I_Attr := Attr;
  I_Length := NrMax;
End;

Function BS_Intr.ReadString : String;
Const
  Terminator : Set Of Char =
[Chr(0),Chr(9),Chr(10),Chr(12),Chr(13)];
  Set1 : Set Of Char = [' '..'~'];
Var Tasta : Char;
  PosAct, Inceput : Integer;
  EdString : String;
  a : Byte;
Begin
  BS_String.Draw;
  PosAct := X + Length(Str);
  Inceput := PosAct;
  X := Inceput;
  ReadString := '';
  EdString := '';
  GotoXY (PosAct,Y);
  a := TextAttr;
  TextAttr := I_Attr;
  Repeat
    Tasta := ReadKey;
    If Tasta = #0 Then Begin
      Tasta := Readkey;
      Case Tasta Of
      'K':
        If PosAct > Inceput Then
          PosAct := PosAct - 1;
      'M':
        If PosAct < Inceput+I_Length Then
          PosAct := PosAct + 1;
      End;
    End
    Else
      If Tasta In Set1 Then
        If PosAct < Inceput + I_Length Then Begin
          EdString := Copy (EdString,1,PosAct-Inceput)+
            + Tasta +
            Copy (EdString,PosAct-Inceput+1,
              Length(EdString));
          PosAct := PosAct + 1;
        End;
    GotoXY (PosAct,Y);
    Str := EdString;
    BS_String.Draw;
  Until Tasta In Terminator;
  ReadString := EdString;
  TextAttr := a;
End;
End.
{Listing 4. BS_Tools - ultima varianta }
```

```
Program List16;
Uses Crt,BS_Tools;

Var C : BS_IntrPtr;
  Attr : Byte;
  S : String;

Begin
  ClrScr;
  New (C,Init(12,12,'Introduceti, va rog : ',12,111));
  S := C^.ReadString;
  C^.Move(20,20);
  Dispose (C,Done);
End.

{Listing 5. Cu BS_Tools introducerea de siruri devine simpla}
```

Protecţie prin parolă

O parolă eficientă ar trebui să conţină mai multe caractere, să nu fie prea simplă şi să fie schimbată la intervale regulate. Nume, date de naştere sau combinaţii "fără sens" de regulă nu ridică probleme deosebite în faţa unor "hackeri" împătimiţi. Acronimele sînt recomandabile, ale unui vers sau cîntec binecunoscut. De exemplu:

u(n')	m(ielule)
t(e)	?
d(uci)	l(a)
t(u)	p(ăşune)
,	,
	d(omnule)

în listingul Unlock.asm, această secvenţă e folosită pentru a "deszăvorî" tastatura. Secvenţa e prelucrată, rezultatul prelucrării fiind un cod ce se memorează în parolă. Dacă aţi greşit cumva vreun caracter la tastarea parolei, puteţi oricînd reiniţializa tastînd <ESC>. Din valoarea compilată, este foarte greu de a deduce o secvenţă care să o genereze (inclusă în listing numai din comoditate). Unlock.asm e folosit ca program pereche cu driverul rezident Lockdrv.asm. Acesta e cel care "zăvoreşte" tastatura, nepermiţînd nici un acces la ea pînă ce nu se rulează Unlock. Cum cea mai frumoasă parolă nu face doi bani dacă fişierul iniţial de comenzi, Autoexec.bat, poate fi întrerupt cu <Ctrl-C> sau <Ctrl-Break>, zăvorîrea tastaturii din Config.sys, face imposibilă intruziunea cuiva care nu ştie parola. Desigur, Unlock ar trebui inclus în Autoexec.bat - altfel, nici Dvs. nu mai aveţi acces la tastatură...Aşa cum vă e prezentat, Lockdrv.sys poate fi folosit de fapt ca un cadru pentru a transforma programe rezidente în memorie şi care preiau anumiţi vectori de întrerupere, în device-drivere. Avantajul metodei: prin modul în care un device-driver se integrează în sistem, el ocupă puţin loc. Dezavantajul: spaţiul ocupat nu mai poate fi eliberat. Compilarea se pote face de ex. cu comenzile

```
tasm lockdrv
tasm unlock
tlink /t /x lockdrv, lockdrv.sys
tlink /t /x unlock
```

(I.F.)

```
1   |  title  #unlock.asm
2   |  ; inceput program
3   |  code            segment  word
4   |                  assume   cs:code, ds:code
5   |                  org      100h
6   |  start:          push  bx
7   |                  push  cx
8   |                  push  dx
9   |                  push  ds
10  |                  push  es
11  |  ; deschide canal pentru #Lock#
12  |                  mov   ah,3dh
13  |                  mov   al,0
14  |                  mov   dx, offset DEVname
15  |                  int   21h
16  |                  mov   DEVhan, ax
17  |  ; citeste 8 bytes din #Lock#
18  |                  mov   ah,3fh
19  |                  mov   bx, DEVhan
20  |                  mov   cx, 8
21  |                  mov   dx, offset INT1buf
22  |                  int   21h
23  |                  push  ds
24  |                  mov   ax,2510h
25  |                  lds   dx, INT1buf
26  |                  int   21h
27  |                  pop   ds
28  |                  push  ds
29  |                  mov   ax, 2516h
30  |                  lds   dx, INT2buf
31  |                  int   21h
32  |                  pop   ds
33  |                  mov   ah, 3eh
34  |                  mov   bx, DEVhan
35  |                  int   21h
36  |                  mov   dx, offset message
37  |                  mov   ah, 9h
38  |                  int   21h
39  |  ; initializare
40  |  init:           mov   dx,0
41  |  ; numarul de taste
42  |                  mov   cx, count
43  |  ; introducere scancod-uri
44  |  next:           mov   ah, 0h
45  |                  int   16h
46  |                  cmp   ah, 1
47  |                  jz       init
48  |                  add   dx, ax
49  |                  ror   dx, 1
50  |                  loop  next
51  |                  cmp   dx, pass
52  |                  jnz   init
53  |                  mov   ax, 3h
54  |                  int   10h
55  |                  pop   es
56  |                  pop   ds
57  |                  pop   dx
58  |                  pop   cx
59  |                  pop   bx
60  |                  mov   ah, 4ch
61  |                  int   21h
62  |  ; zona de date
63  |  INT1buf         label    dword
64  |  INT1bufo        dw       ?
65  |  INT1bufs        dw       ?
66  |  INT2buf         label    dword
67  |  INT2bufo        dw       ?
68  |  INT2bufs        dw       ?
69  |  DEVhan          dw       ?
70  |  DEVname         db       "#Lock#",0
71  |  count           dw       11          ; numarul de taste
72  |  ;parola: "utdt,m?lp,d"
73  |  pass            dw       7938h
74  |  message db "keyboard is locked - please unlock it ...$"
75  |  code            ends
76  |  end             start
77  |
```

UNLOCK.ASM

```
1   |  title lockdrv.asm
2   |  ;request-header-structure
3   |  ; pointer to request header
4   |  RQptr      equ   ds:[bx]
5   |
6   |  RQhdr      struc
7   |             db      ?
8   |             db      ?
9   |  ; command code
10  |  cmd        db      ?
11  |  ; status
12  |  stat       dw      ?
13  |             db      8 dup (?)
14  |             db      ?
15  |  ; adresa de transfer
16  |  adr        equ  this dword
17  |  adro       dw      ?
18  |  adrs       dw      ?
19  |  ; contor
20  |  cnt        dw      ?
21  |  RQhdr      ends
22  |
23  |  ; start programm
24  |  code       segment   word
25  |             assume    cs:code
26  |             org   0h
27  |  ; device header
28  |  ; pointer to next driver
29  |             dd      -1
30  |  ; attribute for char-driver
31  |             dw      8000h
32  |  ; strategy routine
33  |             dw      strat
34  |             dw      inter
35  |             db      "#LOCK# "
36  |
37  |  ; data
38  |  RQadr      label  dword
39  |  RQadro     dw      ?
40  |  RQadrs     dw      ?
41  |  INTptr     dw      ?
42  |  INT1bufo   dw      ?
43  |  INT1bufs   dw      ?
44  |  INT2bufo   dw      ?
45  |  INT2bufs   dw      ?
46  |  count  dw  ?
47  |  ; new interrupt routine
48  |  newint:    iret
49  |  strat      proc  far
50  |             mov   RQadro, bx
51  |             mov   RQadrs, es
52  |             ret
53  |  strat      endp
54  |
55  |  ; rutina de intrerupere rezolva de fapt sarcinile
                  driver-ului
56  |  inter      proc  far
57  |             push  ax
58  |             push  bx
59  |             push  cx
60  |             push  dx
61  |             push  ds
62  |             push  es
63  |             push  si
64  |             push  di
65  |             push  bp
66  |             lds   bx, RQadr
67  |             mov   al, RQptr.cmd
68  |             cmp   al, 0
69  |  black:     je        init
70  |             cmp   al, 4
71  |             je        read
72  |  ; eroare: comanda ineacceptabila
73  |             mov   ax,  8103h
74  |             mov   RQptr.stat, ax
75  |  exit:  pop   bp
76  |             pop   di
77  |             pop   si
78  |             pop   es
79  |             pop   ds
80  |             pop   dx
81  |             pop   cx
82  |             pop   bx
83  |             pop   ax
84  |             ret
85  |  inter      endp
86  |  ; citirea
87  |  read:      les   di, RQptr.adr
88  |             mov   cx, RQptr.cnt
89  |             jcxz  exit
90  |             les   di, RQptr.adr
91  |             mov   si, INTptr
92  |             mov   al, cs:[si]
93  |             stosb
94  |             inc   INTptr
95  |             inc   count
96  |             cmp   count, 8
97  |             jne   no
98  |             mov   count, 0
99  |             mov   INTptr, offset INT1bufo
100 |  no:        mov   ah,1
101 |             mov   RQptr.stat, ax
102 |             jmp   exit
103 |  ; initializare driver
104 |  init:      mov   ax, cs  ; permisa o singura data
105 |             mov   ds, ax
106 |             assume ds: code
107 |             mov   word ptr [black], 9090h
108 |             mov   ax, 3510h
109 |             int   21h
110 |             mov   INT1bufo, bx
111 |             mov   INT1bufs, es
112 |             mov   ax, 3516h
113 |             int   21h
114 |             mov   INT2bufo, bx
115 |             mov   INT2bufs, es
116 |             mov   count, 0
117 |             mov   INTptr, offset INT1bufo
118 |             mov   dx, offset newint
119 |             mov   ax, 2510h
120 |             int   21h
121 |             mov   dx, offset newint
122 |             mov   ax, 2516h
123 |             int   21h
124 |  ; sfirsit parte rezidenta #LOCK#
125 |             mov   dx, offset init + 0fh
126 |             mov   cl, 4
127 |             shr   dx, cl
128 |             mov   ax, cs
129 |             add   dx, ax
130 |             lds   bx, RQadr
131 |             mov   RQptr.adro, 0
132 |             mov   RQptr.adrs, dx
133 |             mov   ah, 1
134 |             mov   RQptr.stat, ax
135 |             jmp   exit
136 |  ; sfirsit program
137 |  code  ends
138 |  end
```

LOCKDRV.ASM

```
1  |  unit comuzica; {$D-,L-,S-}
2  |  interface uses dos,crt;
3  |  type tiptabnote = array[0..32759,0..1] of byte;
4  |  var nrnote:word; tabnote: ^ tiptabnote;
5  |  procedure cintec;
6  |
7  |  implementation
8  |  var salv:pointer;indice,contor,indnot:word;
9  |
10 |  const valnote:array[1..84] of word =
11 |      ( 65, 69, 73, 78, 82, 87, 92, 98,104,110,116,123,
12 |      131,139,147,156,165,175,185,196,
                              208,220,233,247,
13 |      262,278,294,312,330,350,370,392,
                              416,440,466,494,
14 |      524,556,588,624,660,700,740,784,
                              832,880,932,988,
15 |      1048,1112,1176,1248,1320,1400,1480,1568,
                              1664,1760,1864,1976,
16 |      2096,2224,2352,2496,2640,2800,2960,3136,
                              3328,3520,3728,3952,
17 |      4192,4448,4704,4992,5280,5600,5920,6272,
                              6656,7040,7456,7904);
18 |
19 |  procedure tratimp; interrupt;
20 |  begin
21 |      if contor = 0 then
22 |      begin
23 |          nosound;
24 |          indice: = (indice + 1) mod nrnote;
25 |          indnot: = tabnote ^ [indice,0];
26 |          if (indnot0) and (indnotU) then
27 |              sound(valnote[indnot]);
28 |          contor: = tabnote ^ [indice,1] and 127
29 |      end;
30 |      dec(contor)
31 |  end;
32 |
33 |  procedure  cintec;
34 |  begin
35 |      if nrnote > 0
36 |      then
37 |          begin
38 |          getintvec(28,salv); indice: = nrnote-1;
39 |          contor: = 0; setintvec(28,@tratimp)
40 |          end
41 |      else
42 |          begin
43 |              nosound;
44 |              setintvec(28,salv)
45 |          end
46 |      end;
47 |
48 |  end.
49 |
```

MUZICA.PAS

```
1  |  program mtest; uses muzica;
2  |  const note:array[0..29,0..1] of byte =
3  |  ((3,8),(3,8),(4,8),(5,8),(5,8),(4,8),(3,8),(2,8),
4  |   (1,8),(1,8),(2,8),(3,8),(3,12),(2,4),(2,16),
5  |   (3,8),(3,8),(4,8),(5,8),(5,8),(4,8),(3,8),(2,8),
6  |   (1,8),(1,8),(2,8),(3,8),(2,12),(1,4),(1,16));
7  |  begin
8  |      nrnote: = 30; tabnote: = @note; cintec;
9  |      (* orice secventa de instructiuni *)
10 |      write('Tastati Enter pentru terminare');
11 |      readln; nrnote: = 0; cintec
12 |  end.
```

MTEST.PAS

Vă place muzica ?

Prezentăm în continuare un exemplu de utilizare în Turbo Pascal a întreruperii de timp 28 (1Ch). Acest exemplu ilustrează posibilitatea de a executa în paralel două acţiuni: - prima, funcţiile propriu-zise ale programului; - a doua, derularea unei melodii în paralel cu execuţia programului (ceea ce există la majoritatea programelor de joc).

În tabloul VALNOTE (programul MUZICA.PAS) sînt date frecvenţele pentru următoarele note: DO RE MI FA SOL LA SI DO.

În sursa program MTEST.PAS:

În tabelul NOTE este dată partitura. Fiecare notă este precizată prin două numere: primul este indicele în tabelul de frecvenţe, al doilea este durata notei în unităţi de 55 ms.

Bibliografie: Sistemul de operare DOS. Ghidul programatorului - editat de asociaţia Micro-Informatica Cluj-Napoca.

mat. Ioan Cozac

Ştergere rapidă

Deseori, se doreşte ştergerea completă a unei dischete, fără însă a o reformata . Pentru aceasta, trebuie şterse toate fişierele din toate subdirectoarele - ba probabil printre ele mai sînt şi fişiere protejate la scriere, care trebuiesc întîi "deprotejate" pentru a putea fi şterse. Cu programul QFORMAT, o dischetă deja formatată poate fi rapid "reformatată". Programul şterge pur şi simplu cele două tabele FAT precum şi directorul rădăcină, după care suprascrie toate sectoarele dischetei. După ce s-a operat o astfel de ştergere, tot ce a fost pe dischetă este definitiv şi iremediabil dispărut. La startare, programul întreabă în ce unitate se va lucra (A sau B). Alte unităţi sînt blocate; o ştergere accidentală a harddisk-ului este exclusă. Funcţional, programul este echivalent cu WipeDisk din setul de utilitare Norton.

(I.F.)

```
1  |
2  |  Const Maxsec = 124;
3  |  Var Buf:Array[1..512*MaxSec] of Byte;
4  |  Sector, Drive, Count, i : Word;
5  |  Boots, Fats, FatSecs, DirSecs, Rest: Word;
6  |  ch: Char;
7  |
8  |  Procedure ReadSectors;
9  |  Function RSector: Boolean;
10 |  begin
11 |      InLine ($55/$a1/Drive/$8b/$0e/Count/
12 |          $8b/$16/Sector/$bb/Buf/$cd/
13 |          $25/$72/$04/$b0/$01/$eb/$02/
14 |          $b0/$00/$9d/$5d/$88/$46/$ff);
15 |  end;
```

```
16 |  begin
17 |  if Rsector = False then begin
18 |      Writeln ('Eroare la citire !!!');
19 |      Halt (255);
20 |  end;
21 |  end;
22 |
23 |  Procedure WriteSectors;
24 |  Function Wsector: Boolean;
25 |  begin
26 |      InLine ($55/$a1/Drive/$8b/$0e/Count/
27 |            $8b/$16/Sector/$bb/Buf/$cd/
28 |            $26/$72/$04/$b0/$01/$eb/$02/
29 |            $b0/$00/$9d/$5d/$88/$46/$ff);
30 |  end;
31 |  begin
32 |  if WSector = False then begin
33 |      WriteLn ('Eroare la scriere !!!');
34 |      Halt (255);
35 |  end;
36 |  end;
37 |
38 |  begin
39 |  WriteLn (' *** Stergere discheta ***');
40 |  repeat
41 |      Write ('Unitatea (0 ptr. A, 1 ptr. B): ');
42 |      ReadLn (Drive);
43 |  until (Drive = 0) or (Drive = 1);
44 |  Write ('Sterg intr-adevar ? (d/n): ');
45 |  ReadLn (ch);
46 |  if UpCase (ch) = 'D' then begin
47 |      Sector : = 0;
48 |      Count : = 1;
49 |      ReadSectors;
50 |      Boots : = Buf[15] + Buf[16]*256;
51 |      Fats : = Buf[17];
52 |      FatSecs : = Buf[23] + Buf[24]*256;
53 |      DirSecs : = (Buf[18] + Buf[19]*256) div 16;
54 |      Rest : = Buf[20] + Buf[21]*256;
55 |      Sector : = Boots;
56 |      Count : = FatSecs;
57 |      ReadSectors;
58 |      for i : = 4 to FatSecs*512 do Buf[i] : = 0;
59 |      for i: = 1 to Fats do begin
60 |          WriteSectors;
61 |          Sector : = Sector + FatSecs;
62 |      end;
63 |      Count : = DirSecs;
64 |      for i : = 1 to DirSecs*512 do Buf[i] : = 0;
65 |      WriteSectors;
66 |      Sector : = Sector + DirSecs;
67 |      for i: = 1 to 512*MaxSec do Buf[i] : = 241;
68 |      Rest : = Rest-Sector;
69 |      while Rest  0 do begin
70 |          if Rest  MaxSec then Count : = Rest
71 |          else Count : = Maxsec;
72 |      WriteSectors;
73 |      Sector : = Sector + Count;
74 |      Rest : = Rest-Count;
75 |  end;
76 |  WriteLn ('S-a sters totul de pe discheta !');
77 |  end;
78 |  end.
79 |
80 |
```

QFORMAT.PAS

ChDir confortabil

Mai ales la structuri de directoare stufoase, trecerea dintr-un director într-altul este deseori o operaţie care consumă timp şi efort din partea utilizatorului. Programul CHDIR.PAS uşurează această operaţie. La lansare, i se dă ca parametru o parte a numelui din calea la care ar trebui să se ajungă. În continuare, programul parcurge toate subdirectoarele de pe unitatea curentă care conţin în nume şirul specificat ca parametru. Dacă se găseşte un astfel de subdirector, numele complet al căii respective este afişat pe ecran. Dacă acesta este subdirectorul căutat, se apasă <RETURN> şi schimbarea de cale se efectuează. La acţionarea oricărei alte taste, se caută alte subdirectoare care se conformează criteriului specificat.

(I.F.)

```
1  |  {$M 64000, 0, 0}
2  |  uses Dos, Crt;
3  |  Var Ds: DirStr;
4  |      Ns: NameStr;
5  |      Es: ExtStr;
6  |  procedure ChangeDir (path: string);
7  |  var S: SearchRec;
8  |  begin
9  |      FindFirst (path + '*.*',$12,S);
10 |      while DosError = 0 do begin
11 |          if (S.Attr and $10 = $10) and
12 |              (S.Name[1] < > '.')
13 |          then begin
14 |              if Pos(Ns + Es, S.Name) >  0 then begin
15 |                  GotoXY (1, WhereY); ClrEol;
16 |                  Write ('Move to: ', path + S.Name);
17 |                  if ReadKey = #13 then begin
18 |                      Chdir (path + S.Name);
19 |                      Halt(0);
20 |                  end;
21 |              end;
22 |              ChangeDir (path + S.Name + '\');
23 |          end;
24 |          FindNext (S);
25 |      end;
26 |  end;
27 |
28 |  begin
29 |  Fsplit (FExpand (ParamStr (1)), Ds, Ns, Es);
30 |  Ds : = Ds[1] + ':\';
31 |  ChangeDir (Ds);
32 |  GotoXY (1, WhereY); ClrEol;
33 |  Write ('No directory change!');
34 |  end.
35 |
```

CCD.PAS

Poşta redacţiei

"... noi avem nevoie de foarte multă literatură de specialitate. Citind o carte de teoria proiectării sistemelor informaţionale (apărută la o editură românească, bineînţeles) am ales o listă de denumiri de cărţi de care aş avea nevoie în primul rînd. Lista o anexez, subliniind poate cele mai necesare cărţi."

Ovidiu Ciubotă, str. Florilor 6/2, apt. 35, 277068 Chişinău, Republica Moldova, URSS

1. Ionescu C., Sabău Gh., Cojocaru A., Avram V. - Baze de date relaţionale, Lito A.S.E., 1985
2. Drăghici M. - Îndrumar metodologic pentru proiectarea schemei conceptuale a bazei de date, ICSIT-TCI, 1986
3. Bara I., Râureanu M. - Experiment privind arhitectura unui sistem inteligent de gestiune a bazelor de date relaţionale, ICSIT-TCI, 1986
4. Gh. I. Pisău - Elaborarea şi implementarea sistemelor informatice
5. V. Penescu ş.a. Fişiere, baze şi bănci de date
6. Pescaru V., Catona I., Duţă I., Popescu C., Stratan I. - Fişiere, baze, bănci de date, Ed. tehnică 1976
7. Dumitrache V., Unghianu M. - Băncile de date, Ed. ştiinţifică şi enciclopedică, 1982
8. V. Baltac, A. Davidoviciu, C. Maşec ş.a. - Sisteme interactive şi limbaje conversaţionale
9. P.Constantinescu ş.a. - Sisteme informatice, modele ale conducerii

Ne-ar părea tare rău ca această scrisoare să treacă prin conştiinţa noastră ca un simplu fapt divers. Este - şi trebuie să fie - mult mai mult. îi rugăm pe toţi cititorii noştri să recepţioneze acest apel - chiar dacă scrisoarea a venit pe adresa noastră. Şi nu numai de la cititorul nostru (care se întîmplă să fie primul nostru corespondent de dincolo de Prut). Ci de la toţi cei ca el. Faptul că speranţele cititorului nostru au găsit, prin intermediul revistei, o adresă concretă, nu ne scoate din anonimat. Poate că cineva se va apuca să facă o treabă organizată. Dar să nu aşteptăm - sîntem convinşi că vor fi bine utilizate toate cărţile ce vor fi trimise. Direct la Chişinău (preferabil!), sau întîi nouă.. Cei care consideră că pot trimite cărţi de specialitate, sînt rugaţi să o facă. Mulţi dintre noi au cărţi ce pot desigur prezenta interes - şi de care ne putem dispensa.. Sigur că unele sînt învechite şi nu mai au valoare - dar există şi multe cărţi care îşi păstrează valoarea în timp. Nu cred că există multe biblioteci de întreprindere care să nu aibă cărţi pe care s-au aşternut multe degete de praf. Şi care ar putea prezenta o reală valoare pentru colegii noştri de dincolo de Prut, prin simplul fapt că tratează nişte probleme de specialitate în limba română. Să fie chiar imposibil ...? Puţin sceptici - sincer, nu credem că va fi extrem de multă muncă de prestat - ne oferim şi noi ca 'mediatori'. Cărţile ce se vor aduna ca donaţii la noi promitem să le ducem într-un fel sau altul la cei cărora le sînt destinate. Şi nu vedem nici un motiv ca printre cei care fac donaţii să nu se afle şi edituri, instituţii, întreprinderi.

Apropo de instituţii - o trecere în revistă a fişierului nostru de abonaţi relevă un fapt puţin surprinzător: marea majoritate a celor care au făcut abonamente la revista noastră sînt persoane particulare. Cîteva firme particulare, cîteva (puţine) Centre de calcul, cîteva instituţii de stat - şi atît! Dragi cititori,, ştim că uneori este incomod, şi că uneori biblioteca întreprinderii la care lucraţi funcţionează prost sau nu funcţionează de loc. Dar mai ştim şi că sînt foarte puţini directori sau contabili şefi care să citească revista noastră. Neavînd solicitări exprese în acest sens de la subalterni, nici unul dintre aceştia nu va face abonamente la ilustra necunoscută revistă If! Iar bibliotecara - dacă există - este, probabil, cel puţin încurcată dacă trebuie să decidă dacă merită sau nu să facă abonament. Sigur că revista e scumpă - sîntem conştienţi de acest lucru, singura alternativă ar fi să renunţăm. Şi nu vrem să ne dăm bătuţi. O variantă posibilă ar fi creştera tirajului - dar pentru aceasta, difuzarea ar trebui să se întîmple cu totul altfel decît se întîmplă de fapt. Nu putem ieşi din cercul vicios în care ne învîrtim - decît cu ajutorul D-voastră!

Revista e scumpă pentru că tirajul e mic; tirajul e mic pentru că riscul nedifuzării corecte, laolaltă cu preţul ridicat al hîrtiei şi tipăririi, nu ne permit pur şi simplu să ieşim într-un tiraj mai mare. Abonamentele sînt o certitudine - cu ele în spate, tirajul poate creşte. Şi dacă abonamentul este scump - poate că pentru întreprinderea sau biroul în care lucraţi nu este totuşi un efort semnificativ!

Nu vrem acte de caritate - dacă nu merită să supravieţuim, foarte bine, vom dispare, ca atîţia alţii. Poate am calculat greşit, poate că ne-am întins mai mult decît ne este plapuma, poate că pur şi simplu nu ne pricepem.

Dar dacă revista vă place - cît de puţin - atunci vă rugăm să faceţi acel mic efort de a ne face cunoscuţi şi altora. Noi facem revista - dar am vrea ca ea să fie un pic şi a Dvs. De fapt, fără Dvs., ea nu ar mai avea sens.

Al Dvs.,
ing. Iosif Fettich

Wanted !! Reward!!

Am primit bani pentru abonamente de la următoarele persoane sau adrese, care au uitat să ne comunice adresa, respectiv numele:

- MX Consulting Group Bucureşti
- Gliga Jenica, Cluj
- Nemeş Corina
- Miron Cornelia, Arad
- Sibiu, str. Bîrsei nr. 10, bl.39, ap.17

Cine îi cunoaşte, e rugat să îi determine să ia legătura cu redacţia, pentru a completa datele lipsă.

Mica publicitate

Vînd HC-85, şah-computer Mephisto, joystick-PC, unităţi floppy 3,5", dischete 3,5", socluri Tulipe, diferite componente. Negrea Dan, str. 9 mai, nr. 3, sc. A, apt. 13, 5500 Bacău, tel. 931/44450.

Ofer: jocuri, utilitare şi documentaţii pt. calculatoare Spectrum. Szentes Attila, str. Pescarilor, bl. 6, sc. A, apt. 9, 4400 Bistriţa.

Cumpăr microdrives cartridges (cartuşe cu bandă) pentru calculatorul Sinclair QL sau Spectrum şi software pentru QL. Tel. 90/754782.

În numărul viitor:

- **10 programe antivirus**
- **DTP şi fonturi**
- **DOS extendere**

Important !

Doriţi să aveţi garanţia că nu veţi pierde nici un număr din "if" ? Încheiaţi un abonament folosindu-vă de talonul alăturat, sau adresaţi-vă difuzorilor noştri autorizaţi:

Baia Mare
Omni Computer
(Cudalbu Octavian)
tel. 36433

Brăila
Teodorescu Constantin
tel. 47203

Bucureşti
Dracea Tudor
tel. 210340

Cluj Napoca
AbMod
- Magazin Hardware-Software
b-dul 22 Decembrie nr. 135
- Sediul central
tel. 111090

Craiova
Stătescu Paul
tel. 48117

Deva
Toma Tudor
tel. 10144, 10234

Oneşti
Brumă Cornel
tel. 24222-236

Oradea
AbMod
tel. 60278, 44476

Rădăuţi
Burciu Florin
tel. 63483

Sighişoara
Beer Mihai
tel. 73638

Timişoara
Tomoroga Mircea
tel. 77422

Căutăm difuzori/distribuitori autorizaţi
pentru difuzarea revistei "**if**" !.
Condiţii avantajoase !
Rugăm persoanele interesate să ia legătura cu redacţia.

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 172, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if**, pe o perioadă de |_____| apariţii, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 57 lei. Preţul unui abonament se va calcula cu formula: 57 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

if 6 / 91 **Data: ..... Semnătura. .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

Lista faxurilor autorizate de Ministerul Comunicaţiilor pentru a fi conectate în reţeaua naţională de telecomunicaţii

Listă aprobată la 5 iunie 1991

ADLER ROYAL FX 835
AEG OLYMPIA OLIFAX 330
AGORIS 200 EGT
ALCATEL 3602
ALCATEL CT200
ALTECHAF 2500
AMEFON AF 1000
AMCO TFX 200
AMSTRAD FX 5000
AMSTRAD FX 9600 T
ARFAX 2000
ASCOM FAX 150
ATFAX 9600 FAX CARD
AUDIOVOX AFX 1000
AUDIOVOX AFX 2000
AUDIOVOX AFX 3000
AVATEX EFAX 88
AVATEX EFAX 110
BOS BTT 1010
BROTHER 210
BROTHER 305
CANON FAXPHONE 8
CANON FAXPHONE 15
CANON 80
CANON 110
CANON 120
CANON 230
CANON 260
CANON 270
CANON 270S
CANON 350
CANON 410
CANON 450
CANON 510
CANON 520
CANON 710
CANON 730
CANON 750
CITIZEN JFX 1961
DATA FAX 300 SERIES
DEFAX 2 DEVELOP
DIAFAX 001 SF 1000
EFAX EF 110
EPSON Priority FAX 1000
FAX LINE
FAXLINE 3090 PHILIPS
FUJITSU DEX 80
FUJITSU DEX-TEN 3125
FUNAI FA-X2000
GECO UA 2100
GOLDSTAR LF110
GOLDSTAR LF 330
GOLDSTAR LF 510
HARRIS 3M 111
HARRIS 3M 115
HARRIS 3M 2110
HIFAX 17E
INFOTEC 6110
INFOTEC 6112
INFOTEC 6500
INFOTEC 6510
INFRAFAX 9600SF
ITEC FN 10
ITI IF 510
ITT 3531
KONIKA 120
KONIKA 150
KONIKA UBIX FT500
LANIER 115 AD
LANIER 2225
MERCEDES X9
META QX 101
MINOLTA 250
MINOLTA 160/161
MINOLTA 360/361
MITA TC30
MITA TC40
MITSHUBISHI FA60Z
MURATA F 1
MURATA F 30
MURATA M 900
MURATA M 1100
MURATA M1-1600
MURATA M 1800
NASHUA F810
NASHUA INFOTEC 6500
NEFAX 2 NEC
NEFAX 10
NEFAX 11
NEFAX 18
NEFAX 310
NEFAX 4100F
NEFAX 4500
NIPPON NP 90
NISSEI FAX 53
NISSEI FAX 303
NISSEI FAX 305
NISSEI FAX 320
NITSUKO FXE500
OLIVETTI OFX 325
OLIVETTI OFX 530
OKIFAX OF7
OKIFAX OF8
OKIFAX OF8 M
OKIFAX OF17 H
OKIFAX OF17 L
OKIFAX OF38
OKIFAX OF110
PACKARD BELL PB FAX 100
PACKARD BELL FAX 400
PANAFAX 135
PANAFAX UF120
PANAFAX UF130
PANAFAX UF150
PANAFAX UF160
PANAFAX UF250
PANAFAX UF270M
PANAFAX UF400
PANAFAX 420
PANASONIC KX F50
PANASONIC KX F80
PANASONIC KX F90
PANASONIC KX F90 B
PANASONIC KX F100
PANASONIC KX F120B
PANASONIC KX F3550BS
PHILIPS 3050
PHILIPS 3060
PHILIPS 3090 FAX LINE
PHILIPS 3100
PITNEY-BOWES 7100
PITNEY-BOWES 7800
PITNEY-BOWES 8050
PITNEY-BOWES 8000
POLY FOCUS PF 2200
PRIORITY FAX 2000
REGMA TC 20
REX ROTARY 6003 TC
RICOH FAX 07
RICOH FAX 08
RICOH FAX 20
RICOH FAX 25
RICOH FAX 60
RICOH RF 860
ROADSTAR SF2010
SAGEM SAFAX LAUREATT
SAMSUNG SF1000
SAMSUNG FX1010
SAMSUNG 2010
SAMSUNG SF2020
SANFAX 3
SANFAX 225
SANFAX 330
SANYO SANFAX 100
SCHNEIDER FAX SPF-100
SCHNEIDER FAX SPF-101
SEL FAX 10
SEL FAX 20
SEL-ITT 3532
SEL-ITT 3534
SHARP FO 50
SHARP FO 215
SHARP FO 220
SHARP UX 50
SHARP UX 101
SHARP UX 110
SHARP UX 160
SHARP UX 191
SHARP 1800
SIEMENS HF 2021
SIEMENS HF 2301
SIEMENS HF 2302
SIEMENS HF 2303
STABOFAX 10
STEMARK STF 23
TANDYFAX 1000
TELECOPYMAT 200
TENOFAX 3 TELENORMA
THOMFAX 3605 ALCATEL
TOSHIBA 3400
TOSHIBA BD 7610
TOSHIBA TF111
TOSHIBA TF112
TOSHIBA TF152
TOSHIBA TF211
TOSHIBA TF251
TRIUMPH ADLER FX 905
TRIUMPH ADLER TA FX 910
UNDERWOOD UDF 71
UNIFAX 1412 NEC
UTAX F130
VALIDIAN PF 505
XEROX 295
XEROX 7007
XEROX 7009
XEROX 7010
XEROX 7020
ZICOM F900
ZX 1896 MODEM FAX

"Echipamente INFIA S.R.L."

realizează

Dispozitiv pentru scriere / ştergere memorii EPROM

Comenzile se primesc pe adresa:
Cod 76.600, O.P. Bucureşti 75, C.P. 75-37
Relaţii la telefon: 90-86.45.53 după ora 19,00

Societatea de Informatică **DELTA-X S.R.L.**
Oradea, P-ţa Independenţei Nr.47 A3/23
Telefon: 991-34257, 51804

Vă oferă la preţuri avantajoase următoarele produse program generalizabile pe calculatoare compatibile IBM PC XT/AT:

1. Bal-Con - evidenţa contabilă generală

2. Mi-Fi - evidenţa şi calculul amortizării mijloacelor fixe

3. Ge-Ma - gestiunea şi evidenţa contabilă a valorilor materiale (materiale, obiecte de inventar, produse finite).

Produsele enumerate, sînt puternic parametrizate avînd un înalt grad de generalizare şi au strînse legături între ele, utilizînd colecţii comune de date, transferînd informaţii între ele. Evidenţa contabilă Bal-Con, reprezintă "vîrful piramidei" care pune la dispoziţia celorlalte produse planul de conturi contabile, respectiv primeşte date (note contabile generate automat) din partea acestora. Sistemul este deschis, nefiind obligatorie utilizarea produselor program enumerate împreună, respectiv este facilitată conectarea altor aplicaţii la evidenţa contabilă (salarii, desfacere, etc.). Utilizarea şi operarea este facilă prin meniurile, mesajele explicative concrete apelabile în toate fazele de prelucrare, şi autodocumentare.

Prin programele de instalare livrate, utilizatorul poate configura produsele la caracteristicile echipamentului utilizat.

Versiunile finalizate ale celor trei produse menţionate reflectă legislaţia economică şi reglementările metodologice în vigoare şi vor fi aliniate şi în viitor la eventualele schimbări legislative sau metodologice care vor apare.

4. dAccess - accesare multiuser a bazelor de date compatibile dBASE.

Permite culegerea, prelucrarea şi vizualizarea datelor în regim multiuser, de la 1 la 8 terminale, conectate pe căile seriale COM1 -COMB şi de la tastatura/monitorul calculatorului.

Oferă în regim multiuser, prin taskuri independente, majoritatea prelucrărilor permise de sistemele tip dBASE, definirea acestor taskuri făcîndu-se într-un mod simplu, familiar utilizatorilor.

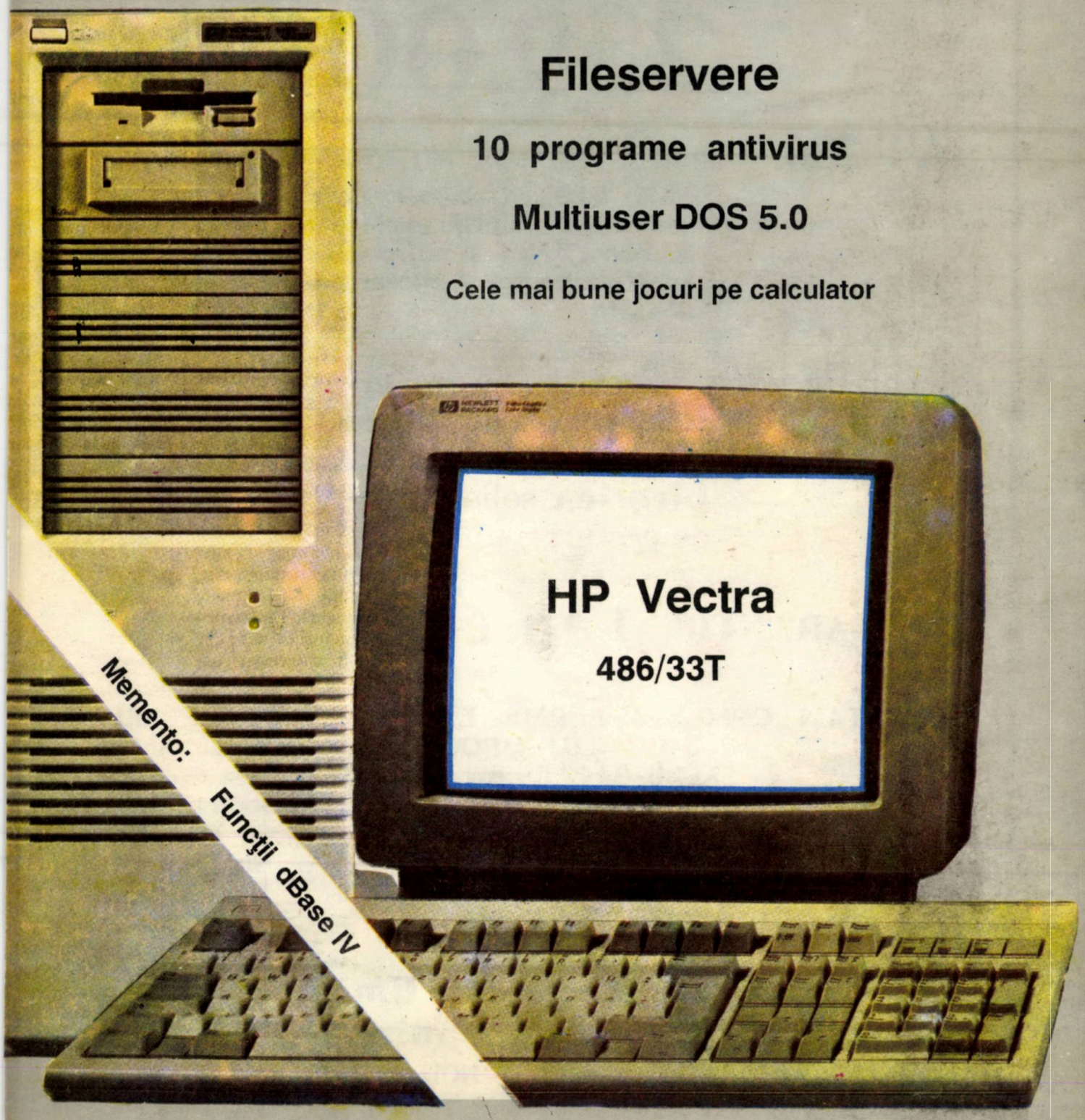
Avînd în vedere, că majoritatea utilizatorilor de tehnică de calcul din economia noastră, dispun de terminale simple în bună stare de funcţionare (VDT-52S, etc.), prin utilizarea acestui produs achiziţionarea unor calculatoare IBM-PC compatibile în configuraţia obişnuită (2 căi seriale) nu exclude utilizarea eficentă a acestor terminale devenite disponibile.

if

CALCULATOARE PERSONALE

Revistă lunară editată de Micro ATCI S.R.L. Tîrgu Mureş

7 / 91
(9)
69 Lei
ISSN 1220-1529



Fileservere

10 programe antivirus

Multiuser DOS 5.0

Cele mai bune jocuri pe calculator

HP Vectra 486/33T

Memento: Funcţii dBase IV

o NOU o NOU o NOU o NOU o NOU o NOU o NOU o NOU o

actiune EGA la

AbMod !

SISTEME COMPATIBILE IBM/PC la numai

239.000 lei

AT 286 12 MHz, 1 MB RAM, HD 40 MB, 1 unitate floppy 1.2 MB 5.25", 1 unitate floppy 1.44 MB 3.5", Monitor color EGA, 1 Mouse, 1 set dischete (10 buc) 5.25", 1 set dischete (10 buc) 3.5", 1 interfata seriala, 1 interfata paralela, tastatura 101 taste

ATENTIE ! Pentru comenzi en-gros firma acorda reduceri substantiale.

Pentru alte echipamente de **calcul** si **birotica**, solicitati lista noastra completa de preturi.

GARANTIE 1 AN DE LA LIVRARE !

ASTEPTAM COMENZI FERME LA UNA DIN ADRESELE:

3400 CLUJ-NAPOCA MAGAZIN AbMod
B-dul 22 Decembrie Nr.135 Tel: 95/156350
3400 CLUJ-NAPOCA Sediul central AbMod
Str. Motilor Nr.11 Tel, Fax: 95/111090 Telex: 31445
3700 ORADEA Str. Dacia Nr.40
Tel: 991/60278 991/44476 Fax: 991/56881

NU UITATI ! Stocul este limitat, si oferta este valabila pina la epuizarea stocului

o NOU o NOU o NOU o NOU o NOU o NOU o NOU o NOU o

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:

ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:

ing. Vasile Brînduş
ing. Attila Darvas
ing. Alin Flaidăr

Tiparul: tipografia Tîrgu Mureş

Revista apare lunar.

Preţ: 69 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Adresa şi telefonul redacţiei:
Micro ATCI, redacţia "if",
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel. 954/31660 (direct),
33612, 24158, 20057, 33511,
int. 134 sau 189
41417 sau 46885 (după ora 19)
fax 954/36489 (Intertours)
telex 65354 (Intertours)

Panta rhei

Totul curge, dar apa trece, structurile rămîn şi cum nu ştim, (şi nu vrem a-nvăţa), să umblăm cu palma întinsă, şi neavînd abilitatea de-a ne pune sub oblăduirea puterilor momentului, nu vă putem întîmpina nici de această dată cu un zîmbet liniştitor.

"Trăim în miezul unui ev aprins" în care în continuă schimbare nu este doar geografia politică ci şi geografia informaticii. Firme care pînă mai deunăzi se băteau pe viaţă şi pe moarte, pentru acapararea pieţii, devin aliate. I.B.M. şi Apple au semnat un contract de colaborare, (detalii în pag. 4), care va modifica sensibil relaţiile în lumea "celor mari". În afara proiectelor comune şi a faptului că vor păşi împreună pe piaţă, în viitor se va miza pe microprocesoare Motorola, şi pe platforme soft comune, ceea ce fără îndoială este o lovitură puternică pentru Intel şi Microsoft.

La închiderea ediţiei am aflat şi că firma Ashton-Tate a fost preluată de Borland, iar firma Digital Research a fost preluată de Novell. Ce ne va mai aduce viitorul, şi ce implicaţii vor avea aceste schimbări?

Altfel ce mai faceţi? Bine, sănătoşi? Calculatoarele pe care lucraţi nu se comportă ciudat? Nu aveţi musafiri nepoftiţi? Dacă da, citind articolul de prezentare a 10 programe antivirus, puteţi alege antidotul cel mai potrivit. Dacă există!

Pentru ne-"liniştea" Dvs., vă informăm că zilele trecute am văzut la lucru primul virus autohton. Dacă virusul este activ şi se tastează întîmplător combinaţia de litere "Jos", calculatorul răspunde afişînd în continuare combinaţia de litere "Iliescu!". Virusul se răspîndeşte cu ajutorul fişierelor cu extensia COM şi corpul lui conţine mesajul "The first Romanian Political Virussian". Puteţi scăpa de el, (de virus), ştergînd fişierele infectate.

Dar să vă dăm şi o veste bună. După, de acum, cunoscutele serii de cărţi de informatică editate de MicroInformatica din Cluj, noi edituri lansează serii de titluri dedicate informaticienilor. Editura Cristian din Bucureşti, a lansat colecţia intitulată "Tehnocrat", prima carte apărută numindu-se "Prezentarea sistemului de operare MS-DOS", DATACOM din Cluj a lansat colecţia "Informatica de buzunar", colecţie din care au văzut lumina tiparului volumele "IBM - PC în lucrări", "AutoCAD" şi "Lotus 1-2-3", iar editura Facla din Timişoara este prezentă în librării cu volumul "Forth - Concept informatic şi limbaj de programare", avîndu-i ca autori pe Radu Berindeanu şi Agota Matekovits.

Începînd din numărul 9/91 al revistei iniţiem un concurs: "1024 lei pentru cel mai bun program sursă de cel mult 1024 Byte". Sînteţi programator? Aţi scris unul sau mai multe programe care rezolvă o anumită problemă în mod genial? Atunci nu ezitaţi, şi trimiteţi-ne programul sursă, împreună cu datele despre compilatorul utilizat şi adresa Dumneavoastră. În fiecare lună vom premia cel mai bun listing cu 1024 lei. Singura condiţie: textul sursă nu trebuie să fie mai lung de 1024 Byte (1 KByte). Listingurile publicate şi nepremiate vor fi onorate cu tariful obişnuit. Vă aşteptăm!

ing. Romulus Maier

SILICON ST
DRIVE IN... BIG LAP 50m
TUNING TUNING!
SECOND HAND COMPUTER
J. CUTTHROAT
158m
CASH & CARRY
V. OBIS 50m
VIRUS
BILL'S BIT-STOP
PLATINEN WASCHANLAGE
F.C. KENNEDY-WALSH
Copy Shop
RENT-A-MAC
HARD WARE CAFE
PLATTEN-WECHSEL IN NUR 5 MINUTEN!
ST. LUCAS '90

Cuprins

Ştiri

- IBM şi Apple împreună pag.4
- dBase cu suport SQL pag.5
- Compilator C OOP pentru Windows pag.5
- Coprocesoarele vor accelera Windows pag.6
- Noi harddisk-uri de 2,5" pag.6
- Dumping pag.6
- Hello CAD-FANS ! pag.6

Magazin

Plăcerea jocului pag.7

Cele mai interesante jocuri pe calculator şi cele mai bune programe utilizator ale anului 1990 au fost premiate la al treilea festival "European Computer Trade Show" din Londra.

Vă prezentăm

10 programe antivirus . . . . . pag. 10

- Antivir Plus
- Virusfinder PC
- Dr. Solomon's Antivirenwerkzeuge
- Iris Antivirus Plus
- Norton Antivirus
- PCcillin
- PC Plus Virus Doctor
- McAfee Scan
- Turbo Antivirus
- Virus Blocker

Fundamente

Multitasking pag.20

În legătură cu apariţia Windows 3.0, tot mai des apare termenul de multitasking. Cîteva din elementele de bază ale acestui concept sînt analizate în acest articol.

Partiţionarea harddisk-ului pag.38

DOS - extendere pag.42

Programarea sub DOS în mod protejat

Sisteme de operare

Digital Research Multiuser-DOS 5.0 . . pag.24

Reţelele sînt prea scumpe pentru multe întreprinderi. Se caută soluţii mai avantajoase - una din ele ar putea fi Multiuser DOS 5.0 .

Reţele locale

Fileserver: alegere corectă pag.26

Dintre toate PC-urile din reţea, fileserver-ele sînt cele mai importante.

Cursuri

OOP - partea a 5-a pag.29

În partea finală a acestui curs sînt examinate problemele care apar la planificarea ierarhiilor de obiecte.

Turbo-Prolog - partea întîi pag.34

Limbajul de programare Prolog este un instrument excelent pentru rezolvarea de probleme logice. În prima parte, a unei serii de 3 episoade, cîteva din principiile de bază.

Hardware

Interfaţa SCSI - Prezentare şi utilizare . pag.46

Practică

- Lupa de timp pag.50
- Tabela ASCII confortabilă pag.50
- Tabela ASCII rezidentă pag.51
- Economie de memorie RAM pag.51
- Igiena zilnică pag.52

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Editorial . . . . . pag. 1
- Poşta redacţiei pag.54
- Mica publicitate pag.54
- Ne-am interesat pentru D-voastră . . pag.53

Lista modemurilor avizate de Ministerul Comunicaţiilor

IBM şi Apple împreună

În lumea calculatoarelor se schimbă întreaga geografie. Cei mai mari rivali IBM şi Apple au încheiat un contract preliminar care prevede cooperarea lor în toată lumea. Împreună cei doi parteneri deţin aproximativ 40% din piaţa mondială de PC-uri.

Şeful firmei Microsoft, Bill Gates a avut o presimţire. "Concurentul se cheamă acum IBM", a anunţat el cu cîteva săptămîni mai înainte, printr-o notă internă, pe toţi membrii conducerii Microsoft. El a presimţit deja de cîteva săptămîni că harta valabilă pînă în acest moment a alianţelor şi a rivalităţilor lumii calculatoarelor va ajunge de nerecunoscut.

Presimţirile nu l-au înşelat pe Gates: IBM şi Apple au anunţat că vor apărea de-acum înainte împreună în faţa întregii lumi. A fost semnat deja un contract preliminar de cooperare în toate domeniile. Contractul dintre IBM şi Apple schimbă cu 180 de grade geografia "cine cu cine împotriva cui".

Cei doi inventatori ai PC-ului, Apple şi IBM, au speriat restul lumii din branşă anunţînd că de acum încolo ei vor coopera, în toate domeniile: dezvoltarea de software, integrarea calculatoarelor Apple-Macintosh în arhitectura calculatoarelor IBM AIX, dezvoltarea procesoarelor RISC cît şi o platformă Multimedia comună pentru industrie.

Contractul de cooperare al foştilor rivali este o ştire extrem de proastă pentru unii dintre colegii lor de branşă: producătorul de procesoare INTEL, care pînă acum aproape că a deţinut monopolul asupra procesoarelor PC va afla din contract că IBM şi Apple îşi vor procura de acum înainte procesoarele de la Motorola. Şi nici Bill Gates nu se va bucura atunci cînd va citi detaliile cooperării IBM-Apple: cei doi parteneri vor dezvolta printre altele şi o nouă "platformă deschisă de software". "Noua platformă" va fi integrată în sistemele de operare existente. Nu se spune nici un cuvînt despre faptul că Microsoft va avea ceva de spus în această dezvoltare.

Înţelegerea

Contractul preliminar dintre IBM şi Apple în vederea colaborării în viitor, pe plan mondial, conţine patru puncte principale. Iată cele mai importante pasaje din declaraţia care a fost dată publicităţii:

1. Joint-Venture pentru software.

Apple şi IBM au de gînd să dezvolte o nouă platformă deschisă de soft de sistem, care să se bazeze pe o tehnologie orientată obiect. Această nouă platformă va cuprinde toate sistemele de calculatoare începînd de la Laptop-uri şi pînă la servere performante. În acest scop se va înfiinţa o firmă de software de sistem, care va rămîne în proprietatea lui Apple şi IBM.

2. Integrarea de Apple Macintosh în sistemele IBM

Ambele întreprinderi vor dezvolta şi vor comercializa produse de reţea şi de comunicaţii. Aceste produse vor mări posibilitatea de introducere a unui Apple Macintosh într-un mediu IBM. În plus ambele întreprinderi vor continua dezvoltarea AIX, care va dispune de suprafeţele utilizator Macintosh şi OSF/Motif.

3. Pieţe semnificativ noi pentru arhitectura IBM-Power-RISC.

Apple are în plan implementarea Single-Chip-ului IBM-RS/6000-Power în calculatoarele Macintosh. Ambele întreprinderi vor să integreze microprocesoarele Power-PC în staţii de lucru şi în Fileservere.

Motorola şi IBM vor să-şi aducă contribuţia la dezvoltarea şi la producţia pe viitor a unei noi generaţii de Power-PC-Chips cu performanţe superioare.

4. Platformă Multimedia comună pentru industrie

IBM şi Apple planifică crearea şi licenţierea în comun a unui mediu software independent de platformă.

Unitatea celor care pierd

Odată cu acest pact cele două puteri ale lumii PC-urilor s-au unit. Atît Apple cît şi IBM au creat cîndva PC-ul. Apple a reuşit deja în 1976, IBM, în schimb, a ignorat noutatea timp de 5 ani şi în 1981 s-a aflat în faţa faptului împlinit că a întîrziat în dezvoltarea platformei anilor 80. Rîvna IBM-ului de a anula distanţa atinsă de Apple a dus la poziţia sa în cadrul pieţii de PC-uri, care a rămas valabilă un deceniu.

Atunci IBM a căutat un furnizor de procesoare PC şi a dat de Intel. Pentru a anula avansul de 5 ani faţă de Apple, IBM a oferit sistemul de operare DOS, creat de Microsoft, spre difuzare şi altor producători. A apărut efectul scontat: cu toate că lumea IBM-DOS a PC-urilor a şchiopătat în urma standardului creat de Apple, industria de calculatoare s-a bazat în întreaga lume pe standardul "deschis" IBM şi a inundat piaţa mondială cu PC-uri ieftine după standardul IBM. PC-ul a fost descoperit a doua oară.

Cu această politică IBM a reuşit să cîştige doar 25% din piaţa mondială la PC-uri, dar scopul principal a fost atins: partea acaparată de Apple a scăzut la 15%. Uriaşul rest al pieţii de calculatoare de 60% şi l-au împărţit între ele clonele IBM-ului. Direct de acest lucru s-a legat şi triumful lui Intel: în afară de Apple nu mai folosea aproape nimeni procesoare Motorola, toţi lucrau cu furnizorul de procesoare al IBM-ului Intel, care şi-a asigurat astfel un cvasimonopol.

Au existat probleme la ambele întreprinderi, care au dus la uitarea vechii rivalităţi. La IBM afacerile se tîrîie, în primul trimestru al anului 1991 a fost înregistrată o pierdere de 1,73 miliarde de dolari. Şeful de la IBM Jone Akers a reacţionat destul de violent vis-a-vis de manageri, punîndu-le în vedere că dacă nu-şi schimbă stilul lent de a acţiona îi va da afară. Şi Apple a avut greutăţi neaşteptate. Tipurile noi "Clasic" şi

"LC" s-au vîndut bine, dacă se însumează numărul de calculatoare vîndute, dar la preţurile joase practicate nici Apple nu s-a putut descurca. Şi conducătorul întreprinderii John Sculley a fost obligat să anunţe concedieri. Astfel, la nevoie, Akero şi Sculley, s-au aliat. Rezultatul tratativelor a fost pactul Akers-Sculley.

dBase cu suport SQL

Se ia dBase IV, se completează cu suport SQL şi se obţine dBase IV-Server-Edition. Împreună cu Serverul Microsoft-SQL el va deveni platforma pentru aplicaţii Client-Server.

Mult aşteptatul "Server Edition" este gata. La 26 iunie Ashton Tate a prezentat cel mai tînăr copil al familiei dBase, în timpul expoziţiei de PC-uri din New York. Server Edition nu este o concurenţă la Oracle sau la SQL-Server, ci un Front-End capabil de SQL. În cadrul aplicaţiilor Client-Server el joacă rolul clientului care filtrează date din serverul bazei de date via SQL. În momentul de faţă Server Edition lucrează doar împreună cu serverul Microsoft-SQL, sprijinirea altor sisteme, ca Oracle sau Sybase, va urma într-o versiune ulterioară.

"Strategia Ashton Tate este de a da utlizatorului acces asupra datelor indiferent de locul în care au fost memorate, sau indiferent cu ce calculator se lucrează, a afirmat David Procter, preşedintele firmei Ashton Tate, în cadrul prezentării lui Server Edition. Primul pas a fost implementarea lui dBase pe cît mai multe platforme începînd cu DOS, UNIX şi pînă la VMS. Urmează deschiderea către serverele de baze de date. Pentru a simplifica trecerea programatorilor dBase de la o platformă la alta, comenzile SQL au fost preluate, în mod transparent, în liniile de comandă dBase. Astfel acelaşi program poate accede atît date locale în format dBase, cît şi interoga baze de date prin serverul SQL. Toate interogările se traduc de Server Edition în SQL, astfel încît este necesară o adaptare la diferitele servere de baze de date. Doar astfel pot fi exploatate funcţiunile specifice ale fiecărui server în parte. Din punct de vedere al programatorului, aceasta înseamnă că aplicaţiile de astăzi vor putea fi adaptate cu minimum de efort la versiunile ulterioare, deoarece Server Edition preia el comunicaţia cu sistemul SQL şi nu trebuie învăţate diferitele dialecte SQL. Dacă se doreşte, un program poate referi direct baza de date prin comanda SENDSQL. Interogările interactive se fac cu ajutorul comenzii SQL SELECT. Utilizatorul poate introduce comenzile direct în timp ce programul se află în execuţie, iar dBase le interpretează în mod dinamic.

dBase-IV-Server-Edition funcţionează pe toate calculatoarele MS-DOS începînd cu 286 cu o memorie de cel puţin 2 MByte. Ashton Tate recomandă însă cel puţin 3 MByte Extended Memory (XMS) pentru a se putea atinge o viteză optimă. Programul sprijină pînă la 15 MByte XMS şi funcţionează şi atunci cînd este pornit din modul 386 al lui Windows 3.0. Poate lucra împreună cu servere sub Novell Netware 286/386, 3COM3 Plus şi Microsoft-OS/2-LAN-Manager.

În SUA dBase-IV-Server-Edition costă în jur de 8000 $ pentru 20 de utilizatori şi poate fi cumpărat deja.

Compilator OOP pentru Windows

Cea mai nouă versiune a compilatorului "Zortech C++ 3.0" şi a bibliotecii "Zink Interface Library 2.0" strălucesc printr-o gamă variată a noilor funcţiuni speciale pentru programarea orientată obiect sub Windows.

Produsele Zortech se numără deja de ani de zile printre primele de pe piaţa compilatoarelor. Cu vesiunea 3.0 a lui C++ Zortech vrea să-şi cosolideze această poziţie. Ea conţine şase îmbunătăţiri, dintre care una în programul propriu-zis: optimizarea codului de program şi astfel o creştere considerabilă de viteză.

Celelalte cinci sînt îmbunătăţiri ale pachetului prin adăugiri: astfel o documentaţie voluminoasă de Windows-API prototipul "Windows Maker", la care Zortech a cumpărat licenţa de la Blue Sky, "WINC Libraries", cu care programele DOS pot fi recompilate pentru Windows şi un 16-Bit-DOS-Extender, pentru care firma a cumpărat licenţa de la Rational System. Astfel cu un comutator al compilator-ului dintr-un program normal DOS se obţine unul pentru Protected Mode.

Întregul pachet a lui Zortech C++ 3.0 va fi oferit pe piaţă în trei versiuni diferite.

Ediţia de bază se pretează pentru DOS şi pentru Windows 3.0. Versiunea dezvoltată utilizează în locul unui 16-Bit-DOS-Extender un 32-Bit-Extender, care consumă doar 15 KByte RAM.

Pachetul tehnico-ştiinţific sprijină în plus coprocesoarele de la Weitek şi specificaţiile "IEEE-748" - de calcul în virgulă flotantă, extensiile "NCEG C" şi mai multe module M++, care pot fi obţinute separat pentru statistică şi alte aplicaţii tehnico-ştiinţifice speciale. Versiunea de bază costă 250 DM versiunea dezvoltată 400 DM şi cea tehnico-ştiinţifică 600 DM.

Foarte intensată a fost şi prezentarea bibliotecii Zinc Interface Library 2.0. Ea lucrează împreună cu compilatoarele firmelor Borland şi Zortech. Astfel programarea aplicaţiilor se poate face în acelaşi timp în 3 moduri: DOS-text, grafic şi Windows 3.0. Codul sursă trebuie mărit cu doar cîteva rînduri de cod suplimentar, pentru a obţine alternativ o fereastră pentru DOS şi alta pentru Windows.

Coprocesoarele vor accelera Windows

Nici Bill Gates n-ar fi visat acest lucru. Branşa calculatoarelor a fost cuprinsă de o adevărată manie Windows.

Aplicaţiile Windows răsar peste tot ca ciupercile după ploaie, foşti opozanţi recunosc deschis aprecierea lor faţă de suprafeţele grafice iar tot mai mulţi producători oferă hardware specific Windows.

În cazul imprimantei laser Windows sau a diferitelor PC-uri Windows (care ar fi create mai ales pentru Windows) este vorba de nişte produse ale unor manageri plini de fantezie, care vor să vîndă tehnică depăşită sub un nou nume.

Lucruri interesante se pot observa în schimb pe ecranele locurilor de muncă CAD din Silicon Valley. Ingineri rafinaţi lucrează aici la coprocesoare care sînt făcute special pentru Windows. Coprocesor nu înseamnă doar "componentă pentru operaţii matematice" cum s-ar putea crede la vederea soclurilor (folosite ce-i drept foarte rar) 287 şi 387 din PC-uri.

Un exemplu bun este cartela XGA de la IBM. Acest cip video posedă exact funcţiunile care iau driver-ului grafic Windows cel mai mult timp: trasarea liniilor, deplasarea blocurilor grafice etc. Mulţi producători vor să urmeze acest exemplu. Weitek, Cyprix şi Intel deja bricolează la cîte un Windows-Turbo-Chip. Ideile lor ajung de la simple cipuri grafice pînă la combinaţii de controlere cache, grafice şi de sunet. În momentul de faţă se pare că la Weitek lucrările sînt în stadiul cel mai avansat.

Inginerii de acolo au prezentat doar de cîteva săptămîni o soluţie cu două cipuri SPARC, care nu conţine doar procesorul, ci şi unitatea de virgulă flotantă, unitatea de management a memoriei şi videocipul. La o integrare corespunzătoare, un cip asemănător ar putea scoate PC-urile din multe situaţii dificile care apar la lucrul cu Windows; mai ales în ceea ce priveşte memoria de lucru şi performanţele grafice.

Şi de la acest procesor ne putem aştepta la accelerarea funcţiunilor matematice, aceasta fiind o specialitate a celor de la Weitek.

Rămîn încă de clarificat cum se va integra cipul Weitek în PC-uri. Unele surse vorbesc despre o cartelă de accelerare Windows, altele de un cip OEM, care va fi integrat pe placa de bază. În acest caz producătorii de PC-uri ar trebui să treacă în fapt la PC-uri speciale pentru Windows.

La Intel toate reflecţiile se învîrt în jurul soclului 487SX. Deoarece după ultimele ştiri 487SX este doar un cip complet 486 cu o modificare foarte mică a configuraţiei de pini, soclul poate fi ocupat şi de un procesor independent.

Acesta oferă pentru viitor posibilităţi interesante de dezvoltare: cine cumpără astăzi un calculator 486SX cu tactul de 20 MHz mîine îl va putea completa cu un procesor de 40 MHz. Căci Intel lucrează, după propriile sale afirmaţii, la cipuri, care funcţionează intern cu o viteză de tact dublă decît BUS-ul. Dau de gîndit şi noile cipuri speciale, cum ar fi 486SX cu cip video integrat sau cu memorie cache mult mărită.

Noi harddisk-uri de 2,5"

Seagate Technologies iese pe piaţă cu două harddisk-uri noi de 2,5" pentru Laptop-uri. Acestea vor fi mult mai rapide - timp mediu de acces sub 20 ms - şi vor avea un buffer de 32 KByte.

Cele două harddisk-uri "ST9077A" şi "ST9038A" oferă o capacitate de stocare netă de 64 MByte şi respectiv 32 MByte şi ambele sînt dotate cu o interfaţă AT. Viaţa medie a celor două harddisk-uri este estimată la 150.000 MTBF.

Dumping

Ministerul comerţului din Statele Unite reproşează producătorilor japonezi Matsushita şi Brother că ar fi vîndut procesoare de texte la preţuri de dumping situate cu 58% sub preţurile "oneste" practicate.

Cei învinovăţiţi neagă aceste reproşuri. Dacă în consfătuirea ce va dura 45 de zile aceste învinuiri nu vor fi ridicate, atunci producătorii japonezi vor trebui să achite nişte amenzi vamale care vor trebui să acopere diferenţele dintre preţul de dumping şi preţul stabilit de minister ca fiind onest.

Hello CAD-FANS !

Numărul revistelor de informatică este în continuă creştere. După PC Magazin, Infoclub , hobBIT, 01 Informatica, salutăm apariţia revistei Hello CAD-FANS, revistă dedicată proiectării asistate de calculator. Spicuim din cuprins: "Ghidul începătorului", "AutoCAD pe felii", "Atelier CAD", "Introducere în AutoLISP. Revista are 32 de pagini şi costă 30 de lei. Pentru detalii suplimentare vă puteţi adresa la: Revista "Hello CAD_FANS", C.P. 39-119 Bucureşti, tel. 53.43.98. Urăm colegilor noştri: "La cît mai multe numere!"

Plăcerea jocului

Cele mai interesante jocuri pe calculator şi cele mai bune programe utilizator ale anului 1990 au fost premiate la al treilea festival "European Computer Trade Show" din Londra.

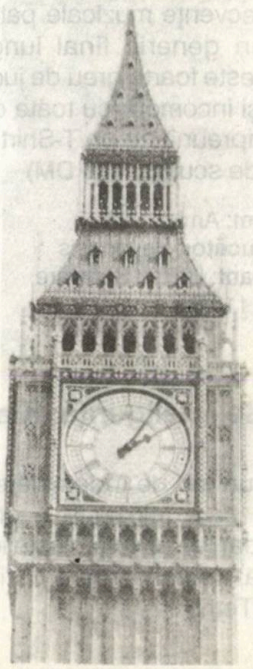
Dacă doriţi jocuri foarte noi nu trebuie să faceţi altceva decît să mergeţi primăvara la Londra la tîrgul european de calculatoare - un tîrg pentru specialişti, care a avut loc doar pentru a treia oară, dar unde se întîlnesc toţi producătorii de hardware şi software, care au ceva de spus în acest domeniu.

În Business Design Center aflat în cartierul Islington au fost prezentate noutăţi extrem de fierbinţi, de exemplu : simulatoare de zbor foarte realiste, care fac să-ţi crească tensiunea (cum ar fi "Birds of Prey" - Electronic Arts sau "The Secret Weapons of the Luftwaffe" - Lucasfilm Games).

Pentru a încerca modul de viaţă în cadrul unor lumi sintetice pe calculator există un demo de Virtual Reality cu cască pentru vizualizare 3D şi cu mănuşă de date.

În 1991 a fost decernat pentru a doua oară la această întîlnire europeană a producătorilor de jocuri premiul "European Computer Leisure Award". Au fost premiate cele mai bune jocuri ale anului cît şi cele mai bune programe utilizator pentru folosinţă privată. Juriul a fost format din redactori de la 32 de reviste de calculatoare din Europa, SUA şi Japonia, avînd în total în jur de 8 milioane de cititori. Produsele au fost premiate datorită superiorităţii lor la categoriile : grafică, sunet, acţiune, originalitate, animaţie, jocuri de aventuri, respectiv cu roluri cît şi jocuri logice, împachetare şi accesorii ("packaging") şi jocuri de simulare foarte apropiate de realitate, jocuri pentru console şi console.

Dintre programele utilizator au fost premiate cele mai bune soft-uri de pictură "Art Package", de realizat muzică "Musik Package", de învăţare "Home Education", în plus



Cîştigătorii	
Jocuri	
Jocul anului:	Lemmings
Cea mai bună grafică:	Shadow of the Biest II
Cel mai bun sunet:	Shadow of the Biest II
Cea mai palpitantă acţiune:	Killing Game Show
Cel mai original joc:	Lemmings
Cea mai bună animaţie:	Dragons Lair II - Time Wrap
Cele mai palpitante aventuri:	Secret of Monkey Island
Cel mai bun joc de gîndire:	Klax
Cel mai bun suport:	Ultima VI
Cea mai bună simulare:	F19 Stealth-Fighter
Aplicaţii	
Cel mai bun program de pictură:	Deluxe Paint III
Cel mai bun program de muzică:	Bars & Pipes
Cel mai bun program pentru învăţămînt:	Fun School 3
Cel mai bun "Home Productivity Package":	Microsoft Works
Cel mai inovativ periferic:	Video Toaster
Cel mai bun pachet multimedia:	Amiga Vision
Home Computer-ul anului:	Amiga 500

perifericul, pachetul multimedia ca şi Home Computer-ul anului (din nou: Amiga 500).

Cele mai multe premii le-a primit casa de soft britanică Psygnosis în calitatea de cel mai bun producător de jocuri al anului cît şi în calitatea de producător al celui mai bun joc pe calculator din 1990 : Lemmings - care este jucat pretutindeni unde se află ecrane Atari ST şi Amiga. În continuare vor fi prezentate produsele premiate.

Psygnosis a obţinut, în clubul de noapte Mega "Hippodrome" de pe Leicester Square, în faţa a peste 500 de oaspeţi internaţionali, alte premii întîi, şi anume pentru jocul cu cea mai bună grafică şi cel mai bun sunet (Shadow of the Biest II), pentru cel mai bun joc de acţiune (Killing Game Show) şi pentru jocul cel mai original (Lemmings).

Cele mai bune programe de jocuri

Cea mai bună grafică: Shadow of the Biest II.

Joc de acţiune, lupte mitice în mediu fantastic (Fantasy-Ambiente), cu un spectacol de grafică color impresionant încă de la început (aşa- numitul "Edel-Intro"), împodobit cu secvenţe muzicale patetice şi cu un generic final lung. În schimb este foarte greu de jucat şi în plus şi incomod (cu toate că se vinde împreună cu un T-Shirt este extrem de scump: 110 DM)

Sistem: Amiga - ST
Producător: Psygnosis
Ofertant: United Software
Preţ: 110 DM

Cel mai bun joc de inteligenţă: Klax

Este un joc de îndemînare plin de idei, care foloseşte o grafică medie, dar ideea de joc este foarte incitantă, asemănătoare cu ideea jocului "Tetris".

Problema : La capătul unei benzi rulante cu 5 coloane se află cîte un cîmp de stivuire. De pe banda rulantă se apropie de jucător blocuri de culori diferite. Cu un fel de lopată jucătorul trebuie să ridice blocurile şi să le depoziteze pe una dintre stive. Stivuirea trebuie realizată astfel încît cel puţin 3 blocuri de aceeaşi culoare să se afle succesiv pe orizontală, pe verticală sau pe diagonală. Dacă acest lucru reuşeşte, aceste blocuri dispar făcînd loc pentru altă tură. În total există o sută de nivele de joc cu grad de dificultate crescător care dau tot mai mult de furcă jucătorului.

Sistem: PC, Amiga, ST, C64, PC-Engine (pentru Mega Drive este în pregătire)
Producător: Tengen, Japonia
Ofertant: Bomico
Preţ: între 50 şi 100 DM (în funcţie de sistem)

Cel mai bun joc de acţiune: Killing Game Show

Este un joc de dificultate medie cu lovituri, pocnituri, cu grafică color destul de bună, cu muzică excelentă, cu efecte sonore foarte bune şi cu vorbire.

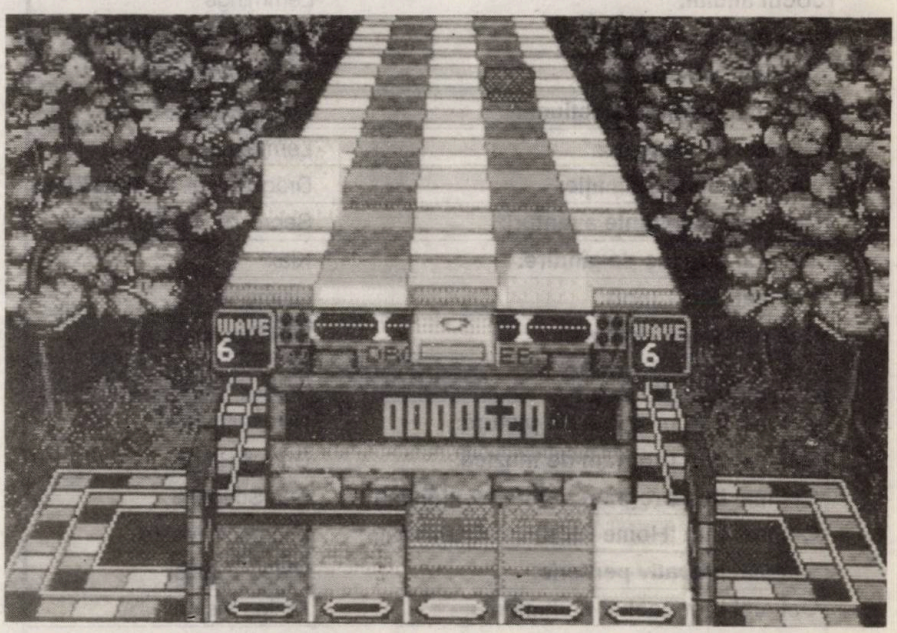
Ca participant la un spectacol de televiziune din viitor, jucătorul este introdus într-un costum zornăitor de robot şi expus într-o zonă plină de primejdii unde trebuie să lupte pentru supravieţuire - ameninţat de mareele mortale de acizi şi de formaţiunile maligne şi răuvoitoare de extratereştri.

Sistem: Amiga (ST în pregătire)
Producător: Psygnosis
Ofertant: United Software
Preţ: 80 DM

Cea mai bună animaţie: Dragon's Lair II Time Warp

Este un joc de acţiune cu grafică bună şi saturat cu sunet, cheful de joc este limitat. Prinţesa Daphne a fost răpită de vrăjitorul cel rău Morlock şi bunul Dirk trebuie s-o salveze. Eroul călătoreşte cu maşina timpului în trecut, în grădina Eden, şi în viitor, mereu în luptă cu Morlock, cu soacra acestuia şi cu cîţiva alţi monştri. Scenele animate pot fi influenţate în momentul oportun de către jucător.

Sistem: Amiga, IC
Producător: Redysoft
Ofertant: United Software
Preţ: 110 DM.



Klax

Jocul de consolă al anului: Tetris

Consola anului: Atari Lynx

Consolele de joc sînt automate de joc, video, de buzunar - computere complete în format Walkman - avînd un ecran de mărimea unei cutii de chibrituri, pe care se pot derula doar jocuri. Soft-ul este sau incorporat sau este livrat în module care se introduc în aparat.

Au ajuns foarte îndrăgite consolele independente de reţea ale specialistului în console Nintendo şi anume Game-Boy şi Atari "Lynx".

Odată cu Game-Boy se livrează jocul de îndemînare Tetris, unul din jocurile care prinde foarte mult, producînd chiar dependenţă.

Preţul - cu consolă cu tot: aproximativ 160 DM. Pînă acuma au fost vîndute 400.000 de bucăţi şi Nintendo ar vrea să ajungă pînă la sfîrşitul lui '91 la 1 milion de bucăţi.

Consola Lynx nu este nici ea mai prejos, şi oferă o veritabilă porţie de Computer High-Tech în format de buzunar, cu monitor color LCD şi cu confort mai mare în deservire decît Game-Boy. În schimb bateria Lynx ţine mai puţin. Ea costă cam 200 DM, şi va fi înlocuită încă în acest an de o versiune mai compactă, mai ieftină şi mai puţin consumatoare de baterii.

Cel mai bun joc de aventuri: Secret of Monkey Island

Este o comedie hazlie şi plină de divertisment folosind o grafică de vîrf (256 de culori din cele 256.000 ale versiunii VGA pentru PC-urile MS-DOS), sunet puternic (sprijină plăcile Adlib şi Soundblastes) şi ritmuri sacadate Calypso şi Reggae.

Acţiune în stil Indiana-Jones: Guybrush Threpwood, care doreşte mult să fie pirat, trebuie să treacă multe probe şi aventuri, pentru a fi acceptat de breasla piraţilor Karibik. Simpatice sînt duelurile cu înjurături în loc de săbii.

Sistem: PC, Amiga şi ST
Producător: Lucasfilm Games, USA
Ofertant: Rushware
Preţ: 100 DM

Cel ami bun suport (Packaging): Ultima IV

Acţiunea în sine este relativ banală - lumea Sosaria trebuie salvată de răii Gargyles, ceva asemănător cu alte jocuri scenice, dar este un joc fantastic realizat într-un mod foarte teatral şi cu o grafică excelentă. Totul se petrece într-un scenariu de ev mediu, un cosmos jucărie cu caractere diferenţiate psihologic, nuanţări morale, toate fiind completate de un inventar admirabil de consecvent şi plin de detalii. Ca supliment se livrează o hartă din pînză, un manual minuscul şi o piatră "magică" - în orice caz marfă multă pentru bani puţini (100 DM).

Sistem: MS-DOS (Amiga şi C64 în pregătire)
Producător: Origin, Texas, USA
Ofertant: Rushware
Preţ: 100

Jocul anului: Lemmings

Este un joc Amiga animat într-un mod deosebit, cu grafică superioară şi sunet deosebit.

Problema: De a conduce poporul Lemmings, cît se poate de nedecimat, peste distanţe cu obstacole cu grad de dificultate tot mai ridicat pînă la "Templul Salvării". Handicapul: Lemingii sînt prostuţi şi îl urmează pe primul dintre ei în orice situaţie - chiar şi la moarte. De aceea este nevoie de un simţ foarte dezvoltat de reacţie, planificare şi gîndire strategică şi tactică - altfel prostuţii cad în prăpastii, vor fi lipiţi de pămînt de o greutate, prăjiţi în foc sau mîncaţi de o plantă omnivoră.

Lemmings a transformat chiar şi scepticii care afirmau: "jocurile pe calculator sînt sterpe", în fani.

Sistem: Amiga, Arari-ST (Versiunea PC în pregătiri)
Producător: Psygnosis
Ofertant: United Software
Preţ: în jur de 80 DM

Cel mai bun simulator: F19 Stealth-Fighter

Problema: De a zbura cu cel mai modern bombardier american de camuflaj în misiuni secrete deasupra zonelor de criză (pentru începători: Libia, pentru aşi: Europa de Est sau zonele potenţiale de război: regiunea golfului şi anume Iran).

Jocul de simulare a zborului şi a loviturilor este destul de macabru, sînt atacate chiar şi uzine atomice. O simulare depăşită din punct de vedere politic (rămasă la nivelul anilor războiului rece), dar foarte realistă din punct de vedere militar, tehnic şi aviatic.

Asemănător cu filmul Top Gun, jocul este aproape perfect şi aduce la suprafaţă instinctul de ucigaş fără mustrări de cunoştinţă pe care îl poate descoperi în sine orice pacifist convins.

Sistem: MS-DOS, Amiga, Atari ST şi C64
Producător: Microprose USA
Ofertant: Rushware
Preţ: 100 DM.

(I.M.)

Dăunătorii calculatoarelor

Viruşii calculatoarelor sînt spaima oricărui posesor de PC. În acest articol vom porni împreună pe urmele celor mai întîlniţi viruşi.

Viruşii calculatoarelor sînt una din cele mai "secretoase" teme ale prelucrării electronice a datelor. Cum trebuie să te comporţi cu ei, şi chiar recunoaşterea existenţei lor, nu este recunoscută ca o temă în sine de mulţi producători hard şi soft. Abia după ce se întîmplă necazul, se reacţionează. Şi totuşi, tocmai această opinie este eronată. Căci altfel cum ar fi putut "viruşii" să-şi capete acest renume de temut.

Spaima corelată cu nestiinţa reprezintă de regulă un bun teren de acţiune pentru afacerişti. Din acest motiv, în ultima vreme, numărul viruşilor calculatoarelor a fost umflat intenţionat. Se poate pune întrebarea, dacă toate programele de sabotaj, căutate de programele antivirus, există şi în realitate. De exemplu se vorbeşte în literatură de virusul "Agiplan" (prezent şi în listele viruşilor "recunoscuţi"), o specie care a fost stîrpită cu siguranţă. El a apărut la sfîrşitul anilor 80, la o singură firmă şi nu a mai fost semnalat nicăieri în altă parte. Numai primii 17 octeţi ai acestui virus au fost recunoscuţi, motiv suficient ca de atunci încoace să se caute prezenţa acestor octeţi. În Africa de Sud, un programator a croşetat în jurul acestor 17 octeţi un nou virus. Cine caută, acela găseşte.

Uimitor este faptul că numai foarte puţini dintre viruşii efectiv răspînditi, sînt programaţi să producă pagube. Majoritatea acestor programele se ţin de farse, pe monitor, la imprimantă sau în difuzor. Şi introducerea datelor de la tastatură pot deveni opera activităţii unui virus. Toate acestea nu justifică însă panica.

Există foarte puţini viruşi care să reprezinte un pericol real pentru utilizator. Cel mai renumit dintre aceştia este virusul "dBase", care manipulează baze de date şi care a fost apreciat ca fiind "cea mai periculoasă lovitură dată prelucrării electronice a datelor". În fapt acest virus a fost semnalat numai de două sau de trei ori. Probabil că acest virus nu mai poate fi descoperit decît în cercurile "cercetătorilor viruşilor".

Gurile rele spun că problema principală vizazi de viruşii calculatoarelor nu sînt viruşii înşişi ci chiar aceşti cercetători. Cu acribia unui pasionat colecţionar de timbre, viruşii sînt mai întîi "descoperiţi", adică: ei sînt pasaţi cercetătorilor şi cît mai urgent sînt analizaţi, clasificaţi şi catalogaţi. Ca urmare a instinctului ancestral al colecţionarului, fiecare modificare a unui virus, chiar şi banală, este clasificată ca "variantă" sau chiar ca "mutaţie". Pentru publicul cu mai puţine cunoştinţe de specialitate, este indiferent dacă, intenţionat sau nu, un virus a suportat o modificare necontrolată. În fapt există tot mai mulţi programatori care, cu mai mult sau cu mai puţin succes, manipulează viruşii astfel încît aceştia nu mai sînt recunoscuţi de programele antivirus.

"Din punctul de vedere al majorităţii tehnicilor neinteligente de căutare a viruşilor, utilizate încă, aceasta este o cutezanţă."

În sine vorbind, un virus care nu mai este semnalat, trebuie să devină neapărat un "nou" virus. Aşa se face că numărul viruşilor catalogaţi trece peste 500. Doar una, pînă la două, duzini din aceste "boli" au însă o răspîndire reală. Aproape ca şi în viaţa reală, căci pojarul este mult mai răspîndit decît elefantiazisul.

Printre primii viruşi s-a numărat şi cel programat de germanul Ralf Burger. Pentru scopuri pur demonstrative, se înţelege: honni soit qui mal y pense. Astăzi autorul, avansat ca "expert în sisteme" şi "popă al viruşilor", în cartea sa "Grossen Computer-Viren Buches" (Marea carte a viruşilor calculatoarelor), preferă să nu-şi mai amintească de acele vremuri, fie din cauza înduioşătoarei simplităţi a programului, fie din cauză că în cercurile de specialitate este considerat ca "agent provocateur". Variaţiunile pe tema lui Burger au devenit între timp legiuni şi sînt desemnate ca "Pixel", "Vienna" sau "Lisabon".

O cunoştinţă veche este şi "170X", unde "X" poate lua valorile "1" sau "4", numărul rezultat dînd lungimea virusului. Una din cele mai noi variante are lungimea de 1714 byte, dar poate fi considerată

Top 25

Cei mai răspîndiţi viruşi în Germania (în ordine alfabetică)

1260	Fish	Perfume
170x	Frodo	Stoned
8 Tunes	FuManchu	Vacsina
April 1st	Ghost	Whale
Burger şi variante	Invader	XA1
Dark Avenger	Jerusalem	Yankee Doodle
DataCrime	Kennedy	Yoshi
Devil's Dance	Music Bug	
Disk Killer	Oropax	

identică cu celelalte. În primele versiuni acest virus lăsa literele de pe monitor "să cadă", fapt care i-a adus numele de "Herbstlaub" (vierme de toamnă), "Cascade" sau "Falling Letters". Unele versiuni distrug şi date prin suprascriere pe harddisk. Dacă aceste versiuni sînt "experimente genetice" produse în laboratoarele "doctorilor" de viruşi, a rămas o enigmă.

Un alt veteran este "Vineri 13", denumit şi "Jerusalem Virus". În zilele de vineri care cad pe data de 13 a lunii, toate programele lansate în execuţie sînt şterse în mod automat. În celelalte zile, pe monitor apare un dreptunghi negru şi calculatorul este "frînat" prin bucle de întîrziere. Şi în acest caz există variante care pot deveni active în alte zile ale săptămînii.

În 1989 a făcut vîlvă virusul cu sonorul nume "Data Crime". Dacă un program infectat este apelat după data de 13 octombrie 1989, pe monitor apare mesajul "DATA-CRIME VIRUS, RELEASED 1ST MARCH 1989". Apoi harddisk-ul este făcut inutilizabil prin ştergerea unor informaţii importante de pe el. Conform unei declaraţii în revista americană "Mondo 2000", în acest caz ar fi vorba despre un virus al mediilor: o nadă aruncată de ziarişti. După cît se spune, se pare că cel puţin în SUA, nu ar fi fost anunţat nici un caz credibil de pagube datorate acestui virus.

Unii viruşi au fost gîndiţi ca mici glume. Astfel aşa-numitul "Stoned-Virus", care se ascunde în sectorul de boot al dischetelor sau harddisk-urilor, afişează pe monitor mesajul: "Your Computer is stoned now. Legalize Marijuana!" Virusul "FuManchu" îşi mărturiseşte opiniile politice interactiv: dacă se tastează numele unui anumit om politic, face nişte comentarii care nu sînt tocmai pentru urechile copiilor.

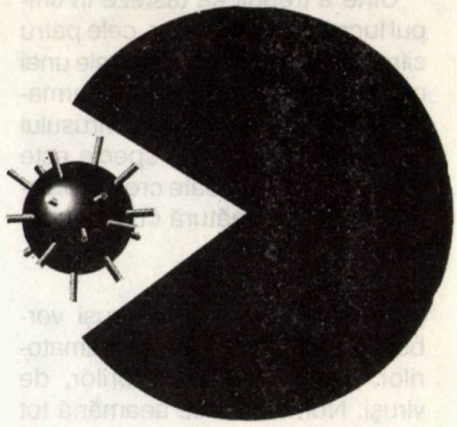
După ce calculatorul este pornit din nou cu o combinaţie de taste, este afişat mesajul: "The world will hear from me again!!!". "Oropax" şi "8 Tunes" enervează utilizatorul cîntînd fals melodii cunoscute, cum ar fi valsul "Dunărea albastră". Şi "Yankee Doodle", cu multiplele sale variante, face parte din această categorie, ca şi "Music Bug", un virus al sectorului de boot din cea mai nouă generaţie.

Ziua de 1 aprilie, Crăciunul, şi alte zile importante îi înspiră de asemena pe programatori. Virusul "1 April", de exemplu, după fiecare infestare reuşită scrie pe ecran: "You have a virus". În data de 1 aprilie este afişat mesajul "Ha Ha Ha It's April 1st-you have a Virus". Virusul "XA1", o prescurtare de la "X-Mas" (Crăciun) şi "A(pril) 1", nu se manifestă numai iarna: de Crăciun este afişat pe monitor un brad de Crăciun cu lumînări. La 1 aprilie, însă, virusul distruge domenii sistem importante.

De o natură mai ciudată este virusul "Joshi" care provine din India. Într-o anumită zi din ianuarie, la pornirea sistemului, el cere: "Type Happy Birthday, Yoshi." Dacă dorinţa îi este îndeplinită, atunci se poate lucra normal mai departe. Probleme pot să apară doar la utilizarea tastaturilor germane, deoarece virusul "nu ştie" că tasta "Y" se află în locul tastei "Z" şi invers.

Dezvoltarea calculatoarelor de la maşini de calcul pure la aparate multimedia este reflectată şi în tematica viruşilor. Virusul "Kennedy", face reclamă formaţiei Punk-Rock "The Dead Kennedys", căreia îi urează "viaţă lungă". Cercetătorii de viruşi, americani, au greşit considerînd că este vorba despre o trimitere la familia Kennedy.

O perplexitate asemănătoare a domnit şi la apariţia virusului "Devil's Dance", care la pornirea "la cald" (warmstart) a calculatorului pune întrebarea "Have you ever danced with the devil under the weak light of the moon?", semnată: "The Joker". Celor cărora li s-a întîmplat să observe virusul, şi care



au pierdut filmul "Batman", le-a fost uşor să creadă că este vorba despre ceva mistic. Ca şi în cazul celorlalţi viruşi în care au fost cifrate cuvintele "line after death" (Phoenix), naturile mai sperioase au putut să creadă că acesta este primul virus care "supravieţuieşte" opririi calculatorului.

Se pare că cea mai plăcută activitate a programatorilor de viruşi este să-i facă pe cercetătorii viruşilor să-şi piardă siguranţa. Astfel există un virus care îşi modifică în permanenţă chipul, numit "Whale", şi a cărei principală sarcină pare să fie de afi cît mai greu de analizat. Ceva asemănător se întîmplă şi în cazul lui "Fish". Şi virusul cu numele "Casper", denumit şi "1260", din cauza lungimii sale, a stîrnit vîlvă în cercurile cercetătorilor, deoarece el nu mai conţine un şir de caractere constant şi devine astfel imun în faţa metodelor tradiţionale de căutare. Ţipătul de spaimă al cercetătorilor nu este justificat, autorului articolului ce a servit ca bază acestei prezentări, Thomas Dehn, nu i-a trebuit mai mult de o oră pentru "a sparge" cifrul lui "1260". Viruşii de acest gen pot fi priviţi ca farse studenţeşti.

O situaţie asemănătoare a fost valabilă şi în cazul virusului "Tequila", care a fost "descoperit" şi catalogat în mai multe locuri din Germania. Şi acest virus este cifrat şi conţine printre altele mesajele: "TEQUILA and BEER forever" şi "Loving thoughts to L.I.N.D.A."

Cine a trebuit să tasteze în timpul lucrului la calculator, cele patru cifre care alcătuiesc numerele unei cunoscute ape de colonie germane (4711), a fost victima virusului "Perfume". Şi această specie este relativ liniştită şi poate crea probleme numai în legătură cu alte softuri.

Numele date noilor viruşi vorbesc despre fantezia programatorilor, dar şi a cercetătorilor, de viruşi. Numele lor se aeamănă tot mai mult cu numele jocurilor pe calculator. Astfel există viruşii "Invade", "Lamer Exterminator" (Amiga) şi "Frodo". Nume ca "Dark Avenger", "Ghost" sau "Disk Killer" sugerează pe bună dreptate anumite pericole. Aceşti viruşi fie şterg date, fie duc la blocarea calculatorului. De remarcat este faptul că "imensele pagube datorate viruşilor calculatoarelor", atît de citate în literatură, cel mai adesea sînt cunoscute numai de la a treia mînă. Unii experţi consideră că este vorba despre "o panică creată voit pentru a facilita afaceri". Probabil că şi în acest caz funcţionează veşnicele legi ale zvonurilor: fiecare mai adaugă puţin senzaţional, şi la a cincea mînă şi cele mai neruşinate exagerări par să conţină un sîmbure de adevăr.

Tehnici de tratament

Viruşii calculatoarelor sînt renumiţi şi temuţi. Există aproape tot atîtea programe care oferă protecţie, cîţi viruşi sînt. Vă vom prezenta în continuare zece dintre aceste leacuri bune la toate.

Cînd se vorbeşte în mijloacele mass-media de calculatoare, adesea dintr-o răsuflare este rostit şi cuvîntul virus. Este vorba despre programe care se instalează în calculator şi care execută anumite acţiuni pe care utilizatorul nu le poate controla. Cei mai temuţi sînt viruşii care infectează programele de aplicaţie şi care şterg sau formatează hardisk-ul.

Programele antivirus promit o protecţie simplă şi sigură împotriva unor astfel de dăunători. Zece reprezentanţi ai acestei categorii de programe, au fost luaţi în vizor şi puşi să-şi măsoare forţele cu peste 1000 de viruşi. Este vorba despre: Antivirus plus, Dr. Solomon's Antivirenwerkzeuge, Iris Antivirus Plus, McAfee Scan & Cleanup, Norton Antivirus, PC Plus Virus Doctor, PCCillin, Turbo Antivirus, Virus Blocker şi Virus Finder.

Antivir Plus

"Antivir Plus" (AV+), al firmei H&B EDV, din Tettnang, este programul care şi-a protejat prin lege denumirea, şi ar fi singurul program de acest fel din Germania care ar avea voie să poarte apelativul "Anitvirus". Acest apelativ este utilizat pentru programele care servesc luptei împotriva viruşilor pe calculatoare. Aceste programe nu lucrează, cum s-ar putea crede, ca un cătravirus, ci localizează viruşii mai cunoscuţi şi/sau îi distrug.

Toate programele antivirus au comun faptul că trebuie să fie în permanenţă dezvoltate, deoarece ele trebuie să fie la zi pentru a putea proteja şi împotriva celor mai noi viruşi. Din acest motiv se lucrează şi la AV+.

Pachetul nu constă dintr-un singur program ci din trei programe de protecţie distincte. Un program rezident în memorie "Avir", care controlează memoria de lucru şi care anunţă utilizatorul în cazul în care sînt efectuate accese nepermise la date.

Apoi există programul care "caută" şi "îndepărtează" viruşii numit "Antivir". Acest duo poate fi înlocuit de un program care se mulţumeşte doar să caute viruşi, numit "Avsearch" şi care este distribuit gratuit de către H&B EDV.

Avir este asemănător veteranului shareware "Flushot+", chiar dacă puterea i-a fost ridicată. El verifică şi sumele de control. Sarcina principală a acestui program este supravegherea sistemului pentru a observa activitatea unor posibili viruşi. Programul TSR dă o multitudine de semnale de alarmă, care în cazul utilizării practice, în birou, pot deveni păguboase. Căci o operatoare care nu face altceva decît să culeagă texte cu un editor, mai ales la prima avertizare, nu poate descifra semnificaţia acestor mesaje.

Motivul acestei conştiinciozităţi exagerate l-ar putea constitui faptul că AV+ a fost conceput nu ca o unealtă pentru lucrul de zi cu zi ci ca o ultimă salvare în caz de necesitate. Astfel acest pachet de programe elimină numai acei viruşi despre care există rapoarte credibile privind apariţia lor pe PC-uri. Înlăturarea lor este însă cu atît mai conştiincioasă. Un alt indiciu despre acribia celor de la HGB este faptul că, în timpul eliminării viruşilor, Antivir indică dacă este posibil ca unul sau altul dintre fişiere să fie mai lung decît originalul, din cauză că anumite erori în programarea virusului (de exemplu la "Vacsina") au dus la suprascierea fără rost a unor octeţi şi cu aceasta au distrus probabil date.

În lupta pentru memoria de lucru" împotriva virusului "CSFR1000", AV+ s-a dovedit a fi un tovarăş de luptă de încredere. Nici un dăunător nu a avut vreo şansă să se copieze în alte programe.

Virusfinder PC

Virusfinder este, după cum afirmă editura DMV-Verlag din Eschwege, cel mai vîndut program în Germania de acest gen. Textul

imprimat pe partea din spate a ambalajului promite "anunţarea" unui posibil virus "în cîteva secunde". Cum diagnostichează programul rămîne secretul producătorului.

Programul este livrat pe o dischetă neprotejată la scriere, împreună cu un manual. Acesta conţine o introducere explicită pe tema viruşi ca şi indicaţiile de operare.

Operarea este destul de greoaie, deoarece ecranul este umplut cu un potop de mesaje. Citirea directoarelor este făcută atît de încet încît ai putea crede că s-a blocat calculatorul. Nici viteza de lucru nu este mulţumitoare.

Cea mai mare lipsă a programului o constituie faptul că lipseşte autoverificarea. Programul poate fi lansat în execuţie şi după ce a fost infectat şi poate servi chiar el la răspîndirea viruşilor.

Principiul de căutare este identificarea unor şiruri de caractere semnificative. Aceste aşa-numite "scancodes" (semnături ale viruşilor) pot fi specificate şi de utilizator. Faptul că în cazul acestei operaţii codurile de scanare existente pot fi suprascrise, şi deci modificate, este o concesie care poate deschide poarta actelor de sabotaj. Înafară de aceasta, cei mai noi viruşi, care se autocifrează, scapă de sub control. Aşa se întîmplă în cazul viruşilor "Omicron", "Fish" şi "Tequila".

Virusfinder are şi slăbiciuni de concepţie. Anumiţi viruşi sînt recunoscuţi doar ocazional. Din 12 fişiere infectate cu "Solano", au fost anunţate ca "bolnave" doar 3. Programul poate fi utilizat pentru căutarea de viruşi, dar ca program antivirus şi-a ratat de puţin scopul.

Dr. Solomons's Antiviren-werkzeuge

Sonorul nume al acestui pachet de programe nu este găselniţa unui expert în marketing ci face trimitere directă la producătorul lui: Dr. Alan Solomon este şeful casei de soft S&S Entreprises din Marea Britanie, specializată în recuperarea datelor. Distribuirea versiunii germane a fost preluată de editura Percomp Verlag din Hamburg.

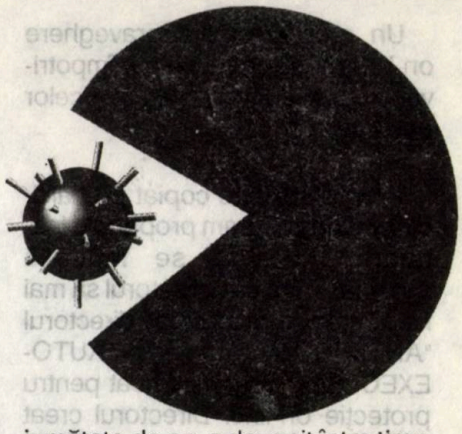
Pozitivă este în primul rînd prezentarea manualului. Utilizatorul este familiarizat cu problematica viruşilor. Cu multe exemple şi observaţii, de asemena în manual fiind prezente şi indicaţii de operare şi fiind explicate problemele care pot să apară la utilizarea pachetului.

Instalarea lui se realizează prin simpla copiere de pe dischetă pe harddisk. Soft-ul constă dintr-un program de căutare a viruşilor şi dintr-o serie de programe de validare a sumelor de control. Operarea este preluată de suprafaţa comună "Toolkit", putînd fu utilizat mouse-ul şi avînd la dispoziţie un sistem de help-uri bine pus la punct.

Deoarece protecţia rezidentă memorie poate fi activată la nivel de program nu este oferită o astfel de protecţie on-line. Ca şi compromis este oferit programul "Guardian" care verifică programele înainte de lansarea lor în execuţie.

La căutarea viruşilor cu programul "Findviru" a fost impresionantă viteza de căutare. Chiar şi mesajul afişat la descoperirea unui virus "infectat cu siguranţă" vorbeşte despre încrederea în sine a programatorului.

Funcţia "eliminare viruşi", punctul slab al pachetului în urmă cu o



jumătate de an, a devenit între timp o funcţie puternică. Chiar şi viruşii exotici, care nu au fost semnalaţi încă în Europa, au fost recunoscuţi şi îndepărtaţi corect.

Posibilitatea oferită, de a căuta "mutaţii" ale unui virus cunoscut, s-a dovedit a fi bine intenţionată, dar n-a avut succes. Nu numai că a fost încetinită căutarea, dar nu s-au găsit nici mutanţi.

Filozofia programului, vizavi de "CSFR 1000" s-a dovedit a nu fi infaibilă. Virusul s-a putut răspîndi copiindu-se în fişierele COM. Abia după o pornire "la rece" a calculatorului, de pe dischetă, programul şi-a făcut corect treaba.

Ca program de căutare şi de îndepărtare a viruşilor Dr. Solomon's Antivirenwerzeuge, poate fi considerat un program de referinţă.

Iris Antivirus Plus

Iris Antivirus Plus este un produs al casei soft israeliene Iris. În Germania el este distribuit de firma Hoffmann din Dusseldorf. Ea a preluat şi suportul tehnic "hotline".

Pachetul Iris constă dintr-o dischetă de 5,25", neprotejată, şi dintr-un manual de cca. 50 de pagini. Acesta descrie modul de funcţionare al programului şi filozofia sa.

Un program de supraveghere on-line oferă protecţie atît împotriva viruşilor cunoscuţi cît şi a celor necunoscuţi.

Programul este copiat pe harddisk cu un program propriu de instalare. Aceasta se întîmplă automat. Fără ca utilizatorul să mai fie interogat, este creat directorul "ANTIVIRUS" şi fişierul "AUTOEXEC. BAT" este modificat pentru protecţie on-line. Directorul creat este protejat la scriere, astfel încît utilizatorii pot avea dureri de cap dacă vor să-l mute în altă parte, sau să-l şteargă (trebuie utilizată comanda "attrib"). Căutarea viruşilor este realizată elegant şi conştiincios. La eliminarea viruşilor cunoscuţi, Iris s-a descurcat bine.

În cursul instalării programului, fişierele sistem (IO.SYS, MSDOS.SYS), ca şi sectorul de boot, sînt copiate pe o dischetă, şi sînt făcute copii şi pe harddisk. În caz de distrugere, ele pot fi reconstituite.

De neînţeles rămîne, de ce Iris preia programele executabile fără o supraveghere generală a sumelor de control. Încredera nelimitată acordată programului de supraveghere s-a dovedit a fi o eroare fatală. În duelul direct cu virusul de test, Iris a constatat infectarea memoriei şi a crezut că a rezolvat eficient îndepăratarea virusului. Ştergerea programului infectat a fost făcută prea tîrziu. La o nouă pornire a sistemului s-a dovedit că interpretorul de comenzi COMMAND. COM era infectat. Ca urmare sistemul s-a blocat. Deoarece era vorba despre un virus necunoscut, nu se ştia unde trebuia căutată cauza: Iris a deblocat cu succes, totuşi, încercarea de infectare.

După o modificare a fişierului "AUTOEXEC.BAT", în care Iris a fost dezactivat, s-a dat ocazia virusului "CSFR 1000", ca în numai cîteva minute, să infecteze nu mai puţin de 112 fişiere. Din cauza absenţei unei funcţii de validare a sumei de control, nu s-a putut stabili care fişiere erau infectate. Utilizatorului nu i-a rămas altă soluţie decît să restaureze în întregime colecţia de programe de pe dischete.

Utilizat împotriva unor variaţii necunoscute ale unor viruşi "vechi", Iris s-a descurcat, dar faptul că programul oferă siguranţă şi împotriva viruşilor care vor veni, şi că deci nu mai are nevoie de actualizări este doar un slogan publicitar.

În ciuda lipsurilor, Iris este un produs stabil. În orice caz, nu strică să fie completat cu un program de validare a sumelor de control.

Norton Antivirus

Numele "Peter Norton" este valabil ca un certificat de calitate în rîndurile utilizatorilor PC. Cu "Norton Antivirus" (NAV), renumita casă de soft îşi face intrarea şi pe piaţa programelor antivirus.

Manualul explică funcţiile şi modul de operare al programului. Introducerea în problematica viruşilor este limitată la remarca: "un virus este un program care determină calculatorul D-vs să reacţioneze altfel decît sînteţi obişnuit". După această definiţie aproape orice actualizare a unui program este un "virus".

Furnitura de livrare conţine două dischete de 5,25" şi respectiv 3,5" şi un manual. Instalarea este realizată automat, la fel ca şi căutarea unor eventuali viruşi. Viteza de execuţie este remarcabilă.

În cazul în care sînt depistaţi viruşi ei pot fi parţial înlăturaţi. Spre deosebire de alte programe, cuprinse în acest test, un fişier infectat cu mai mulţi viruşi nu poate fi curăţat la o singură trecere. Acest aspect, deşi este bine documentat, poate duce, în cazul unor utilizatori cărora le-a fost lene să citească documentaţia, la erori de operare. Norton Computing poate aduce îmbunătăţiri în acest caz.

Protecţia, în continuare, este realizată cu programul rezident în memorie "Interceptor". El necesită cca. 24 KByte RAM. Puterea lui se limitează însă la verificarea infectării programelor cu viruşii cunoscuţi.

În cazul unor noi viruşi, programul s-a făcut remarcat doar prin încetinirea accesului la harddisk. Dubios pare faptul că driver-ul NAV_.SYS al lui Interceptor, a fost modificat, fapt care ar fi trebuit să fie un semn clar al activităţii viruşilor.

Pentru depistarea unor noi dăunători există o funcţie de verificare a sumei de control. Pentru fiecare fişier COM, EXE şi SYS, este depus, în acelaşi director, un fişier cu numele de control, cu atributele sistem "Read Only, Hidden, Sys". Aceasta duce însă la o enormă risipă de spaţiu (cca. 200 KByte pentru 400 de fişiere program).

În plus această funcţie poate produce neplăceri ce nu pot fi înlăturate. Astfel optimizarea harddisk-ului, cu programul aceleiaşi case de soft, "Speed Disk" are sens numai cu ştergerea în prealabil a fişierelor cu sumele de control şi refacerea lor după optimizare. Chiar şi mutarea programelor în alte directoare implică măsuri complexe, deoarece în acest caz cele două sume de control nu vor mai corespunde.

În testul practic, NAV s-a dovedit neajutorat în faţa unor viruşi noi. Nu sînt remarcate nici modificările sumelor de control, nici activităţile viruşilor în memorie şi nici măcar nu sînt semnalate încercările de modificare ale programelor.

Paleta de funcţiuni a programului se limitează la căutarea şi îndepărtarea viruşilor cunoscuţi în SUA. Deoarece multe variante eu-

ropene nu sînt recunoscute, NAV nu asigură o protecţie satisfăcătoare.

PCCillin

PCCillin (PCC) crează impresia unei protecţii profesionale împotriva viruşilor. Din furnitura de livrare, pe lîngă programul însuşi, mai face parte şi un dongle. Acesta este o placă de interfaţă, ale cărui informaţii interne sînt puse la dispoziţia programului, şi fără de care el nu poate funcţiona. El serveşte, după datele producătorului, în primul rînd pentru memorarea unor informaţii despre sectorul de boot.

La o primă instalare, automată, reuşită, conţinutul acestor sectoare este scris în EEPROM (Electrically Erasable Programable Read Only Memory), care este inclusă în dongle.

La pornirea sistemului, programul verifică dacă nu s-a modificat configuraţia memorată la prima instalare. Dacă s-a modificat ceva atunci este afişat un mesaj. PCC trebuie să fie primul program apelat în AUTOEXEC.BAT.

Căutarea viruşilor cunoscuţi, la instalare, are loc cu o viteză acceptabilă, numai că unii dintre cei mai răspîndiţi viruşi nu sînt depistaţi. În plus PCCillin nu îndepărtează dăunătorii din programe.

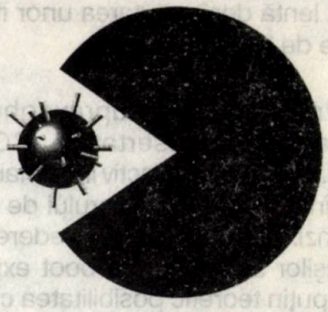
Manualul, pur informativ, face o introducere competentă şi explicită în problematica viruşilor şi lămureşte faptul că PCC este conceput mai puţin pentru căutarea de viruşi cît mai ales ca program de prevenire a infectării cu viruşi. Se porneşte deci de la faptul că "starea normală" a unui PC este fără viruşi şi PCC trebuie să împiedice viruşii cunoscuţi şi necunoscuţi să pătrundă după mecanismele cunoscute şi/sau imaginate în sistem.

Programul rezident în memorie supraveghează toate activităţile sistemului şi în cazul unor activităţi dubioase opreşte sistemul fără a-i mai lăsa utilizatorului nici o şansă. După aceasta sistemul nu mai poate fi pornit decît "la rece" (cu butonul de reset, sau cu comutatorul de alimentare). Metoda se dovedeşte a fi de încredere. Doar proaspătului venit pe piaţă Tequila i-a reuşit să modifice conţinutul harddisk-ului după propria plăcere. Abia la următorul boot, a fost observată modificarea şi a fost refăcută. Dacă această acţiune a fost împiedicată, virusul s-a putut răspîndi în voie. Numai la părăsirea unui program infectat a observat PCC că ceva nu este în ordine. Totuşi este discutabil dacă năzbîtiile celui care a făcut testul pot fi puse în cîrca programului. La instalarea pe un PC infectat apar aceleaşi probleme şi producătorul va trebui să ţină cont de acest lucru.

Supărătoare este înclinaţia lui PCC de a da mesaje de eroare. Pe un 386 programele TSR (Terminate-and-Stay-Resident) şi driverele încărcate în zona de memorie cuprinsă între 640 KByte şi 1 MByte, sînt văzute ca activităţi ale unor viruşi şi sistemul este blocat automat. Acelaşi lucru se întîmplă şi cu compilatorul Turbo-Pascal, care utilizează metode proprii de tratare a memoriei.

Pentru eliminarea acestor efecte secundare, programul poate fi dezactivat cu o combinaţie de taste, dar atunci dispare şi protecţia. Negativ este şi faptul că PCC nu permite stabilirea gradului de infecţie.

Per total PCC asigură o protecţie de nădejde atît împotriva viruşilor cunoscuţi cît şi a celor necunoscuţi. Pentru "dezinfectarea" unor sisteme infectate, nu este nici gîndit, nici indicat. În ceea ce priveşte mesajele de eroare, producătorul mai poate aduce îmbunătăţiri.



PC Plus Virus Doctor

Întreprinderea PC Plus GmbH, din Munchen, a lansat pe piaţă pachetul "Virus Doctor" care constă dintr-un program de căutare a viruşilor, unul pentru îndepărtarea lor şi dintr-un supraveghetor al memoriei numit "Guard". Instalarea se realizează prin simpla copiere a fişierelor pe harddisk. Guard poate fi instalat ulterior, dar se poate şi autoinsera în CONFIG.SYS.

Manualul nu este voluminos, dar este sufucient pentru operarea programului. Temele "Generalităţi despre viruşi" şi "Ce trebuie făcut în cazuri grave?" sînt totuşi prea scurte.

După datele producătorului Virus Doctor este o versiune restrînsă a pachetului realizat în aceeaşi casă Virus Professor, gîndit pentru a oferi ajutor şi protecţie utilizatorilor fără cunoştinţe PC avansate. Programul principal Virus Doctor oferă ca funcţiuni verificarea memoriei, căutarea şi înlăturarea de viruşi ca şi cîteva posibilităţi de instalare pentru Guard. El nu poate însă să construiască sume de control şi nici să creeze un fişier de pornire, care ar permite obsevarea apariţiei viruşilor.

Virus Doctor caută toţi viruşii pe care-i cunoaşte, sau numai pe aceia pe care îi poate şi îndepărta. Căutarea după şiruri de caractere

(scancodes) s-a dovedit sensibil mai lentă decît căutarea unor modele de biţi cunoscute.

Programul de supraveghere Guard este insertat în CONFIG.SYS şi devine activ încă înainte de încărcarea procesorului de comenzi. Din punctul de vedere al viruşilor sectorului de boot există cel puţin teoretic posibilitatea controlării în întregime a sistemului.

În lupta împotriva virusului "CSFR 1000" (unul din cei mai noi), s-a dovedit că acesta a fost blocat cu succes. Nu a trebuit să ne temem de o răspîndire în continuare a acestuia.

Din punctul de vedere al mesajelor Guard prezintă cîteva lipsuri. Modificarea sectorului de boot a rămas necomentată. Tocmai astfel de evenimente nedorite sînt, mai ales la partiţionarea unui hardidsk, un semn de alarmare. Produsul va trebui îmbunătăţit la acest capitol.

Virus Doctor este puţin folositor la recunoaşterea şi îndepărtarea viruşilor. Măsurile de protecţie sînt realizate în primul rînd de programul de supraveghere Guard, care se descurcă mulţumitor atît timp cît nu apar viruşi ai sectorului de boot.

PC Plus a anunţat modificări consistente în program, care să-i crească puterea. Pînă atunci însă programul trebuie să se mulţumească cu un loc în clasa de mijloc.

McAfee Scan

"Viruscan" al lui John McAfee este considerat un program de căutare a viruşilor, pur şi simplu pentru faptul că autorul programului este unul din cei mai cunoscuţi experţi în virusologie. Faptul că este atît de cunoscut nu prezintă însă o garanţie în sine a calităţii programului. Din acest motiv a fost testat şi pachetul său antivirus, livrat ca produs shareware.

A fost testată versiunea 75C a lui Viruscan (Scan). Programul de îndepărtare a viruşilor "Cleanup" purta numărul de versiune 75. În momentul în care apare acest articol va fi disponibilă deja versiunea V77, dacă nu chiar o versiune cu un număr mai mare. Suprinzătoare este frecvenţa cu care este actualizat programul. Producătorul explică aceasta prin faptul că în permanenţă apar noi viruşi pe piaţă, care trebuie recunoscuţi de program şi împotriva cărora trebuie să poată lupta.

În documentaţia livrată pe dischetă se afirmă că sînt recunoscuţi 480 de viruşi. Cifra este exagerată puţin, deoarece parcurgîndu-se cu atenţie lista se pot observa cîţiva viruşi care au fost număraţi de două ori. Mai rămîn totuşi cca. 450 de viruşi.

Înafară de căutarea şi îndepărtarea viruşilor, de cîtva timp, McAfee oferă şi o funcţie de verificare a sumelor de control. Această funcţie adaugă la fiecare fişier executabil 10 Byte în care se înregistrează informaţii pe baza cărora se poate depista ulterior o posibilă infecţie. Dezavantajos este faptul că Scan nu face cunoscut faptul că a adăugat aceste informaţii, utilizatorul putînd interpreta greşit modificarea lungimilor fişierelor respective.

Acest fapt s-a dovedit la prima încercare de a "injecta" programe "curate". Copiile fişierului Command.com de pe lectorul de dischete verificat, după adăugirea sumelor de control au fost recunoscute ca "modificate". Acest fapt nu s-a modificat nici după cel de-al doilea control. Ca şi în cazul anterior, Scan a crezut că a descoperit activitatea unor viruşi. Deoarece şi celelalte programe, din furnitura McAfee, au fost recunoscute ca "modificate" faptul stîrneşte neîncredere. Neîncrederea a sporit atunci cînd Scan a raportat două fişiere nevinovate ca fiind infectate cu "Icelandic" şi respectiv cu "170X". Scuza că programul a depistat codurile de scanare tocmai în aceste fişiere nu funcţionează, deoarece alte programe de căutare a viruşilor nu dau alarme false.

Aceste motive de neîncredere au sporit în cazul programului "Cleanup". Remarca din documentaţie, conform căreia orice virus recunoscut, îl poate şi îndepărta s-a dovedit a fi doar un slogan publicitar. Reparaţiile programelor, în majoritatea cazurilor, constă nu în refacerea programelor infectate ci în ştergerea lor. Cînd programul se apucă de reparat, mai mult strică.

Împotriva virusului "CSFR 1000" programul s-a dovedit a fi neajutorat, desigur pentru faptul că acest virus este nou şi nu este recunoscut de versiunea care a fost testată. Viruscan dă atît de des false mesaje de eroare încît nici cu ajutorul funcţiei de validare a sumelor de control nu se poate stabili răspîndirea infecţiei.

Este de presupus faptul că Scan şi-a asigurat un nume atît de bun, tocmai din cauză că unii utilizatori au luat de bune aceste false mesaje de alarmă şi le-au considerat o dovadă a "conştiinciozităţii" cu care lucrează programul, viruşii anunţaţi nefiind oricum prezenţi. În comparaţie cu partenerii de întrecere, pachetul de programe McAfee mai are mult de recuperat.

Turbo Antivirus

Programul "Turbo Antivirus" al producătorului israelian Carmel Software Engineering, este distribuit în Germania de către firma muncheneză EPG Software. Manualul de 75 de pagini conţine pe lîngă o introducere în problematica viruşilor şi sfaturi pentru operarea în reţea, răspunsuri la întrebările cele mai frecvente, care apar în

asistenţa hotline, ca şi descrierea celor mai cunoscuţi viruşi.

Instalarea se face fără probleme cu ajutorul unui program livrat în pachet. Programul principal "Tntvirus.exe" cuprinde funcţii de căutare şi de îndepărtare a viruşilor, de calculare a sumelor de control ca şi de "imumizare" a fişierelor program.

Surprinzătoarea viteză la stabilirea şi compararea sumelor de control se datorează faptului că în examinare sînt luate numai anumite secţiuni ale fişierelor. Cu toate acestea, după cum afirmă programatorul Eli Shapira, metoda poate determina cu certitudine un posibil atac. Informaţiile cu sumele de control sînt depuse în cîte un fişier pentru fiecare director. Starea sectorului de boot poate fi păstrată intactă, şi la nevoie poate fi refăcută cu ajutorul programului "Bootsafe.exe".

Imunizarea este realizată cu ajutorul unui program care se aşează "ca o coajă" în jurul programului de protejat. Fiecare program astfel imunizat este verificat înainte de lansarea sa în execuţie şi în cazul în care acesta este atacat de un virus se dă un mesaj de eroare.

Ca programe de supraveghere sînt oferite programele "Defender" şi "Tsafe". Defender verifică la încărcarea unui program dacă acesta conţine vreun virus cunoscut. Dacă este depistat ceva, execuţia se întrerupe. Tsafe supraveghează în plus ca şi activităţile virulente ale unor viruşi încă necunoscuţi să fie recunoscute şi întrerupte.

Pentru testul practic, mai întîi a fost lăsat virusul "CSFR 1000" să se răspîndească pe calculatorul de test. Apoi a fost pus la treabă Turbo Antivirus. Programul trebuia să depisteze, pe baza sumelor de control, care fişiere erau infectate. După căutarea în numai patru subdirectoare, programul a raportat că şi-a terminat căutarea şi n-a mai putut fi urnit nicicum pentru a căuta pe tot harddisk-ul. Spre cinstea programului trebuie să mărturisim însă că la majoritatea fişierelor imunizate nu numai că a fost observată modificarea, dar a fost refăcută şi starea lor iniţială.

Programul caută viruşii într-un ritm susţinut. La repararea unor fişiere infectate pot să apară însă probleme: fie se blochează sistemul, fie fişierele nu zînt reparate corect. După afirmaţiile distribuitorului, lipsurile observate vor fi ridicate în curînd.

Virus Blocker

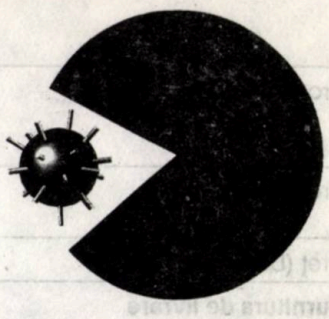
"Virus Blocker" al firmei Expert Informatik din Berlin, este gîndit mai mult ca protecţie în cazul unor atacuri cu viruşi şi mai puţin pentru observarea viruşilor. La dorinţă, programul permite şi blocarea execuţiei unor programe nedorite.

Virus Blocker este livrat împreună cu un dongle, o interfaţă a cărei electronică conţine informaţii fără de care programul nu poate funcţiona. El serveşte în primul rînd ca metodă de protecţie la copiere.

După instalare programul stabileşte starea soft-urilor existente calculîndu-se sumele de control şi memorîndu-le. De neînţeles este faptul că programul nu face mai întîi o verificare automată a sistemului pentru a vedea dacă nu sînt prezenţi deja anumiţi viruşi.

Stabilirea sumelor de control este incomodă. Pentru fiecare tip de fişier (cum ar fi *.COM, *.EXE sau *.SYS) programul trebuie lansat din nou. Din păcate acest lucru se întîmplă şi atunci cînd se doreşte verificarea faptului că anumite fişiere au fost sau nu modificate.

Acest aspect este cu atît mai supărător cu cît producătorul recomandă o verificare zilnică. În plus



Virus Blocker lucrează atît de încet încît utilizatorii ar putea fi tentaţi să întrerupă modificarea. Programul nu permite însă să fie întrerupt. În plus Virus Blocker caută atît de încet viruşii sectorului de boot încît mulţi utilizatori vor fi tentaţi să renunţe la această funcţie.

Ca alte măsuri de protecţie sînt oferite programe care împiedică formatarea suporţilor de date, încărcarea de programe rezidente sau chiar scrierea pe harddisk. Unul din aceste programe permite crearea unui "fişier de logare", în care sînt înregistrate toate apelurile de programe, activităţile lor în memorie, vectorii de întrerupere utilizaţi ca şi modul lor de terminare.

În testul împotriva virusului "CSFR 1000", Virus Blocker a determinat mai întîi căderea sistemului. La o nouă pornire a calculatorului s-a dovedit că fişierul COMMAND.COM era deja infectat. Toate programele .COM apelate din AUTOEXEC.BAT au fost infectate şi ele. Nici după instalarea programului de supraveghere on-line, infecţia nu a putut fi stopată. Fişierele infectate au putut fi localizate după sumele de control modificate.

Rezumînd, Virus Blocker poate fi recomandat, cu anumite rezerve, ca sistem de protecţie. Programul nu excelează în căutarea viruşilor.

(R.M.)

Tabel comparativ

Produs	Antivir Plus	Virusfinder PC	Dr. Solomon's Antiviren-werkzeuge	Iris Antivirus +	Norton Antivirus
Producător	H&BEdv	DMV Verlag	Percomp Verlag	Hoffman	Symantec/ Access
Preţ (DM)	198	149	360	298	295
Furnitura de livrare					
Manual		*			
Dischete					Dischete
Necesar RAM pt. partea rezidentă	14 KByte	9 KByte	4 KByte	22 KByte	24 KByte
Instalare program	Auto	Auto	Auto	Auto	Auto
Mod de lucru					
Elimină	da	nu	da	da	da
Sume de control					
Stabileşte	(Opţiune TSR)	-	47 s	-	140 s
Verifică	(Opţiune TSR)	-	50 s	-	108 s
Caracteristici de operare					
Suprafaţă utilizator grafică	-	*	*	*	SAA
Operare cu mouse	-	-	*	-	-
Documentaţie					
Manual	*	*	*	*	*
Fişier	*	*	*	*	*
Help on-line	-	-	*	-	*
Rezultate de test (pe cei mai răspîndiţi 25 de viruşi)					
Găsiţi	146	123	151	165	148
Indepărtaţi	99	0	140	164	53
Distruse la reparare	0	-	0	0	0
Total					
Găsiţi	926	726	877	912	920
Sume de control					
Stabilire	(Opţiune TSR)	-	47 s	-	140 s
Verificare	(Opţiune TSR)	-	50 s	-	108 s
Căutare pe un disc curat	78	452	42	100	63
Alarme false	-	*	*	-	*
Fişier raport	*	*	*	*	*
Protecţie on-line	*	-	-	*	
Interactiv	*	*	-	*	*
Tip (TSR/Device Driver)	TSR	-	DRV	TSR	DRV
Căutare viruşi rezidentă	-	*	*		*
Poziţie În clasament	4	8	1	2	5

Tabel comparativ

Produs	PCCillin	PC+ Virus Doktor	Scan & Cleanup	Turbo Antivirus	Virusblocker
Producător	Trend Micro	PC Plus GmbH	McAfee Ass.	EPG, Haar	Expert Informatik
Preţ	cca. 380	430	18	298	480
Furnitura de livrare					
Manual			Manual ca fişier		
Dischete					Hardlock-Dongle
Necesar RAM pt. partea rezidentă	14 KByte	11 KByte	37 KByte	16/8 KByte	2 - 10 KByte
Instalare program	Auto	Copiere	Copiere	Auto	Auto
Mod de lucru					
Elimină	nu	da	da	da	nu
Sume de control					
Stabileşte	-	-	405 s	45 s	706 s
Verifică	-	-	405 s	40 s	653 s
Caracteristici de operare					
Suprafaţă utilizator grafică	-	-	-	*	-
Operare cu mouse	-	-	-	-	-
Documentaţie					
Manual	*	*	*	*	
Fişier	*	*	*	*	*
Help on-line	-	-	-	-	-
Rezultate de test (pe cei mai răspîndiţi 25 de viruşi)					
Găsiţi	111	99	149	138	138
Indepărtaţi	0	51	88	125	0
Distruse la reparare	0	0	8	3	0
Total					
Găsiţi	523	844	806	781	856
Sume de control					
Stabilire	-	-	405 s	45 s	706 s
Verificare	-	-	405 s	40 s	653 s
Căutare pe un disc curat	260	130	198	190	310
Alarme false	-	*	*	-	*
Fişier raport	-	-	*	*	*
Protecţie on-line	*	*	-	*	*
Interactiv	-	*	*	*	*
Tip (TSR/Device Driver)	TSR	DRV	TSR	TSR	TSR
Căutare viruşi rezidentă		-	*	*	-
Poziţie în clasament	**9-10**	**6**	**7**	**3**	**9-10**

Multitasking

Cel tîrziu de cînd a apărut Windows 3.0, noţiunea de "multitasking" este în vogă. Pe măsură ce discuţiile avansează, ceaţa se îndeseşte. Iată de ce am considerat necesară explicarea acestei tehnici, precum şi a unor noţiuni ca "preemption" sau "deadlock".

În general, prin "multitasking" se înţelege executarea simultană a mai multor sarcini - "taskuri". Traducerea corectă spune doar că se execută mai multe taskuri - nici o vorbă despre simultaneitate.

Această ambiguitate îşi are sorgintea în faptul că PC-urile uzuale sînt calculatoare cu un singur procesor, deci pot executa la un moment dat un singur program (task). O prelucrare paralelă a mai multor taskuri presupune însă existenţa mai multor procesoare.

Cu toate acestea, prin "multitasking" - la sistemele de operare de pe PC-uri - se înţelege tocmai acest paralelism tehnic imposibil. Şi acest lucru, s-ar putea crede că se întîmplă cînd de sub "Windows 3.0" sau "Geoworks Ensemble" se pot derula mai multe programe aparent simultan. Numai aparent: pe calculatoare cu un singur procesor, multitasking-ul nu este decît o "iluzie optică".

Principiul ce stă la baza acestui efect se poate ilustra cu exemplul unui zidar care lucrează la un şir de case: de la ora 9 la 10 lucrează la casa A, de la ora 10 la 11 la casa B, de la 11 la 12 la casa C, apoi din nou la casa A şi aşa mai departe - pentru spectatorul accidental, cele trei case par să se ridice simultan.

Tocmai această strategie o aplică şi sistemele multitasking. Singura deosebire: schimbarea "locului de muncă" nu se face odată pe oră, ci, de regulă, cam la 0,05 secunde. Ca şi în viaţa reală, şi calculatorului îi trebuie un timp pentru a ajunge de la o casă la alta. Şi chiar dacă sistemele de operare bune reuşesc să facă extrem de repede trecerea, un lucru e sigur: totdeauna, în multitasking, un program se derulează mai încet decît dacă procesorul i-ar aparţine în exclusivitate.

De ce este nevoie de această tehnică pretenţioasă? Uneori, nu se poate altfel. Există, de exemplu, sistemele "multiuser" - constelaţii deci, la care mai mulţi utilizatori, de la terminale diferite, lucrează pe acelaşi calculator. În astfel de cazuri, multitasking-ul este un imperativ - fiecare vrea să ajungă la rînd.

În sistemele cu un singur utilizator în schimb - şi ele formează marea majoritate a PC-urilor instalate - necesitatea multitasking-ului nu mai e deloc atît de evidentă. Bietul utilizator nu poate introduce date cu mîna stîngă, să lucreze cu dreapta în contabilitate şi să mişte în acelaşi timp mouse-ul cu cotul pentru a produce eventual un nou logo al firmei dintr-un program de grafică. Şi într-adevăr, nu este simplă găsirea unor aplicaţii. Cu toate acestea, cîteva există: aşa-numitul "spooler" de exemplu, permite o uşurare semnificativă a lucrului. El face posibilă tipărirea unui document în "bakground", în timp ce se lucrează deja la textul următor. O astfel de funcţie - la fel ca un ceas de exemplu, afişat într-un colţ al ecranului - reprezintă deja o formă simplă de mutitasking: pornind de la un impuls periodic (întreruperea de timer) programul curent este mereu întrerupt. Controlorul procesorului trece atunci la un subprogram. Acesta trimite atunci cîteva caractere de text spre imprimantă sau actualizează afişajul de oră, redînd apoi controlul programului principal.

Astfel de subprograme se găsesc pe mai toate PC-urile, sub forma programelor rezidente (aşa-numitele "TSR"-uri). Marele dezavantaj al acestora constă în faptul că un program întrerupe un altul totdeauna ritmic. Sistemele de operare moderne oferă de aceea un mecanism, în centrul căruia stă un "switcher" (comutator). Acesta permite păstrarea simultană în memorie a mai multor programe, trecerea de la unul la altul făcîndu-se numai la solicitări (prin "hotkey" = "tastă fierbinte", combinaţie de taste sau click pe mouse).

Astfel de exemple dintr-un program de grafică se poate lucra la o ilustraţie, pentru a comuta apoi în prelucrarea de texte şi a o insera acolo. O zonă de memorie comună - numită "clipboard" - face posibil acest transfer de date. Pentru ca acest lucru să funcţioneze, ambele programe trebuie să fie scrise special pentru acest mediu. şi acest lucru din două motive:

- un transfer de date poate funcţiona numai dacă ambele programe "văd" acelaşi format al datelor în clipboard
- ambele programe trebuie să aibă loc simultan în memorie. Deoarece, din motive de spaţiu, deseori acest lucru nu este posibil, este treaba programatorului să pregătească programele pen-

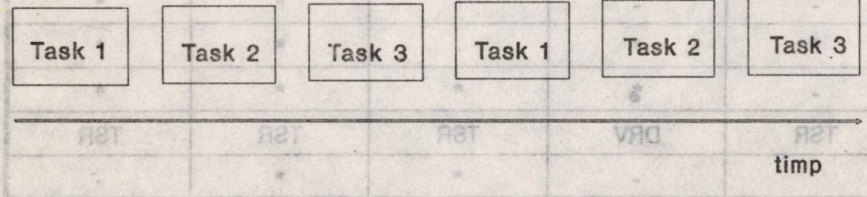


Fig. 1. Fiecărui task îi este alocată aceeaşi cuantă de timp

tru a putea fi rulate în mediul switcher-ului, deci să le "mărunţească" în multe fragmente mici, cărora să le atribuie clasificări de genul "poate fi temporar depozitat pe disc" sau "trebuie să rămînă tot timpul în memorie". Dacă această premisă nu este îndeplinită, trebuie evacuat, respectiv restaurat, întotdeauna întreg programul pe harddisk - ceea ce durează foarte mult.

Cu o prelucrare paralelă - chiar aparentă - acest lucru încă nu are prea multe în comun. Deci între sistemele cu switcher şi cele de multitasking mai există o diferenţă: comutarea de task-uri nu se face "de mînă", ci automat, prin sistemul de operare. Se pune întrebarea: cînd?

Figura 1 prezintă un sistem în care la 3 taskuri li se alocă pe rînd o cuantă de timp identică. Pe cît de onest poate să pară acest lucru la prima vedere, pe atît este de stupid. Un program de prelucrare texte este ocupat mai ales să aştepte tastări de la operator. Oricît de iute ar tasta un operator uman, la timpi de comutare taskuri de ordinul milisecundelor, între două tastări trec mici veşnicii. Deci se risipeşte preţiosul timp al procesorului.

De aceea în general se are grijă ca unui task să i se aloce timp calculator numai atunci cînd nu aşteaptă un "eveniment" - o apăsare de exemplu sau o mişcare a mouse-ului. Premisă necesară pentru acest lucru este desigur ca sistemul de operare să ştie cînd un task aşteaptă un eveniment. Sub Windows de exemplu, această problemă se rezolvă prin faptul că programele aplicative interoghează totdeauna sistemul de operare asupra producerii unui eveniment. Astfel, Windows cunoaşte deja, în mod necesar, o stare a unui program: "în aşteptare". În afară de aceasta, un task se mai poate găsi în stările "pregătit" sau "în execuţie". O diagramă de stări corespunzătoare e prezentată în figura 2. "În execuţie" este programul care tocmai are procesorul,

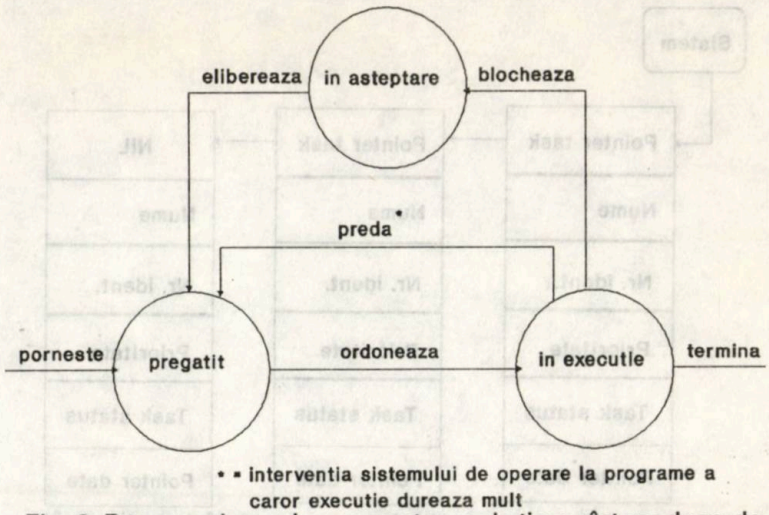


Fig. 2. Programele mari consumatoare de timp sînt readuse de către sistemul de operare în starea "pregătit".

"pregătit" este cel care ar vrea să-l obţină.

Comutarea efectivă a taskurilor o rezolvă o rutină sistem numită "dispecer" (distribuitor).Aceasta opreşte un task şi salvează toate datele sale "personale", toţi regiştri procesorului. Apoi încarcă datele salvate ale următorului task şi îl lasă să continue de unde a fost întrerupt.

Cînd anume intervine dispecerul, este o decizie a altei funcţii de nucleu, pe nume "scheduler". Acest "planificator" decide, care task, cînd ajunge la rînd şi cîte cuante de timp să i se aloce. Decizia se ia cam în felul următor: generatorul de tact al scheduler-ului este deja amintita întrerupere de timer, declanşată de hard de 18,2 ori într-o secundă. Aceasta determină o unitate de timp de 0,055 secunde. O parte din acest timp, scheduler-ul îl consumă el însuşi, deoarece trebuie să se uite dacă unităţile de timp alocate unui task s-au consumat deja. Ceea ce se rezolvă prin intermediul unui contor de ceas. Situaţia devine problematică abia cînd se execută mai multe taskuri. Pentru a păstra privirea de ansamblu, scheduler-ul lucrează de regulă cu liste, care în plus pot fi ordonate după priorităţi. În cel mai simplu caz există numai o singură astfel de listă. Scheduler-ul ştie, care este taskul care tocmai se execută, şi se uită la următorul. Dacă acesta este "în aşteptare", atunci trece la următorul task şi tot aşa mai departe, pînă ce găseşte unul care este "pregătit".

Chiar genială nu este această strategie, deoarece parcurgerea

Procese şi taskuri

Două din noţiunile deseori confundate, relativ la sistemele de operare, sînt cele de "proces" şi "task".

Un task este un subprogram care este "pornit" de un program principal, numit "iniţiator". Prin urmare, el se execută eventual împreună cu alte taskuri în mediul iniţiatorului. Că acest lucru poate duce uşor la conflicte, este evident: procesul, memoria şi alte resurse trebuie folosite în comun.

Un proces în schimb este un program care se poate comporta de aşa manieră de parcă ar dispune singur de procesor. Nu trebuie să ţină seama în nici un fel de alte procese, şi deci poate lansa şi taskuri. Numai resurse precum ecranul sau imprimanta trebuiesc partajate.

Funcţiile de bază ale unui sistem multitasking nu sînt deci task-uri, ci, mai degrabă, procese.

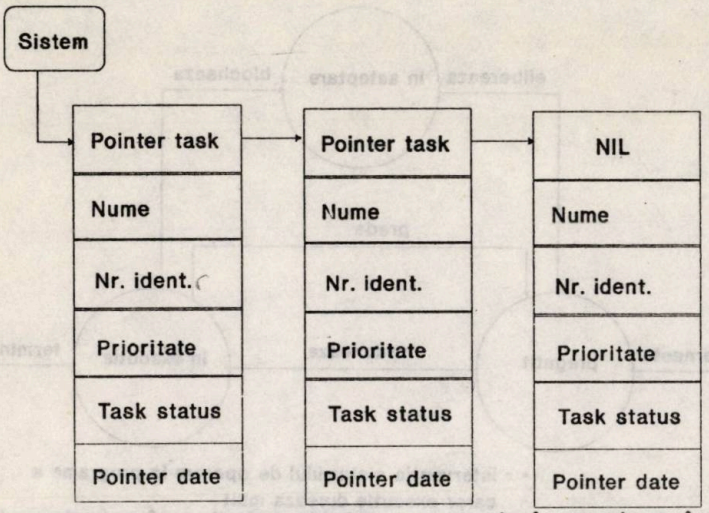


Fig. 3. Toate informaţiile despre un task sînt conţinute în "Task Control Block" (TCB)

listelor costă timp. Aceeaşi pierdere de timp se produce la apariţia unui eveniment: în acest caz, lista trebuie parcursă pentru a afla care este taskul (sau care sînt taskurile) care îl aştepta. Acesta e trecut atunci din starea "în aşteptare", în starea "pregătit".

Pentru a evita aceste întîrzieri, unele sisteme gestionează liste separate pentru taskurile "în aşteptare", "pregătite" şi "active" (ultima fiind interesantă desigur numai în cazul sistemelor multiprocesor). Procedura descrisă se referă la o strategie secvenţială: toate taskurile ajung pe rînd să lucreze, chiar dacă pentru intervale de timp diferite. Cînd s-a ajuns la ultimul, se continuă din nou de la primul - lista de taskuri este o listă circulară. Procedeul poartă numele de "round robin". Nucleul acestuia este aşa-numita "preemption" - dislocare, înlocuire. Un task îl înlocuieşte pe altul, de cum i s-a epuizat timpul alocat.

Contraponderea acestei strategii o reprezintă aşa-numita "strategie de urgenţă". În acest caz, lista de taskuri este sortată după priorităţi. Adică: un task este lăsat să ruleze, următorul porneşte abia cînd predecesorul lui a terminat. În acest caz există două feluri de dislocări (preemption):

- dacă apare un task nou, de prioritate mai mare decît cel în curs, atunci acesta înlocuieşte task-ul care se execută. Există însă şi modele, în care acest lucru nu este permis. În cazul acestora, task-ul este aşezat la coada listei. Algoritmul se numeşte "FCFS" (first come, first served") sau "HOL" (head of line).
- chiar şi într-un sistem ghidat prin priorităţi, un task nu poate fi neîntreruptibil: sistemul de operare intervine regulat, pentru a afla dacă au apărut task-uri noi. Deseori, se ţine şi un fel de statistică, care poate duce la o redistribuire dinamică a priorităţilor. Motivul este simplu: dacă sistemul de operare constată că un task se găseşte mereu în starea "în aşteptare", înseamnă că acesta nu se grăbeşte prea mult. Deci i se reduce prioritatea, şi el se va lăsa înlocuit de alt pretendent.

Nu aşa se întîmplă în sistemele "non-preemtive": sub GEM, Windows 2.1 şi în mare parte şi sub Windows 3.0, în cazul normal se execută un singur task, cel care corespunde ferestrei active. Nimeni şi nimic nu îl poate înlocui - doar dacă task-ul întreabă sistemul de operare dacă nu cumva s-a produs vreun eveniment. De voie, de nevoie, trebuie să facă acest lucru destul de regulat, căci altfel nu află niciodată dacă utilizatorul a selectat un meniu sau a apăsat o tastă. Apelurile de funcţie necesare pentru aceasta redau în mod automat controlul sistemului de operare, care categoriseşte taskul ca fiind "în aşteptare" şi îşi îndreaptă atenţia spre următorul task care poate fi de exemplu ceasul afişat. Acesta s-a interesat dacă s-a mai produs un eveniment tip timer - o întrebare de ceas. Dacă da, e servit cu promptitudine, deci el citeşte ceasul sistemului, desenează noua poziţie a limbilor ceasului pe ecran şi întreabă despre un nou eveniment relevant - moment în care, din nou, controlul trece la sistemul de operare.

Acest procedeu non-preemtiv este considerat deseori un minus (de altfel particula implică lipsa a ceva). Lucrul deseori trecut cu vederea, este simplu fapt că pentru sisteme cu un singur utilizator, această strategie este aproape ideală. Căci utilizatorul individual lucrează de regulă cu un singur program. Şi de ce un program izolat să fie mereu întrerupt de sistemul de operare, numai şi numai ca acesta să constate că nu mai există şi altul care să vrea să ajungă la rînd?

Renunţarea la astfel de întreruperi inutile nu aduce numai avantaje în ce priveşte viteza de execuţie, dar simplifică şi conlucrarea diverselor componente ale unui sistem. Căci chiar neproblematice nu sînt nici întreruperile; orice task se poate afla într-o fază în care să nu poată fi întrerupt. O astfel de fază se numeşte "secţiune critică". Ce anume este critic, este stabilit fie de sistemul de operare, fie chiar de task-ul în cauză. Acesta poate opri, multitasking-ul la începutul unei secţiuni critice printr-o funcţie de genul "**forbid ()**", pentru a-l reporni mai tîrziu cu "**permit ()**".

Sistemul de operare la rîndul lui poate şi trebuie să împiedice accesul simultan a două taskuri la o resursă unică, cum ar fi harddisk-ul sau imprimanta. Imaginaţi-vă că două taskuri tipăresc simultan două texte diferite, fiecare punînd cîte o literă cînd e la rînd.

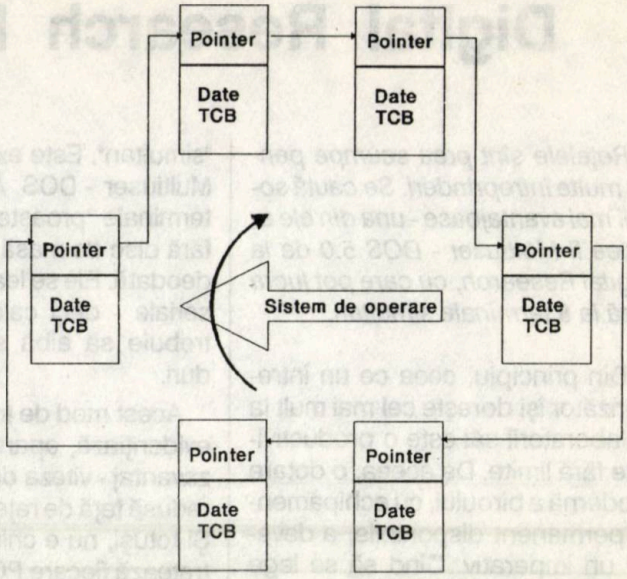
Pentru soluţionarea acestei probleme există două strategii, şi anume excluderea reciprocă ("mutual exclusion") şi accesul exclusiv ("exclusive access"). În primul caz, sis-

temul de operare decide asupra distribuirii drepturilor de acces, în cel de-al doilea, task-ul trebuie să le solicite şi să le predea.

Astfel de solicitări constituie una dintre componentele de bază ale comunicaţiei inter-task-uri bazată pe aşa-numitele "mesaje". În acest context, şi unele rutine trebuie văzute ca task-uri.

De desfăşurarea impecabilă a acestei comunicaţii răspunde sistemul de operare. Că rolul de "regizor" nu este dintre cele mai facile, îl exemplifică aşa-numitul "deadlock" (scurtcircuit): task-ul A aşteaptă un mesaj de la B, iar taskul B aşteaptă un mesaj de la A. Astfel de situaţii duc negreşit la blocarea sistemului şi de aceea ar trebui pe cît posibil evitate sau măcar corectate de sistemul de operare.

Nu aşa se întîmplă sub Windows: aici, întreaga responsabilitate rămîne pe umerii programatorului. El trebuie să aibă grijă ca să nu apară deadlock-uri (blocaje). Ceea ce uneori este dificil, mai ales cînd intră în joc funcţia "Yield ()" ("cedează trecerea"). Aceasta opreşte taskul curent şi porneşte un altul, care deja aşteaptă. Dacă însă un task "cedează trecerea" în timp ce prelucrează un mesaj, pe care un alt task i l-a trimis, se ajunge la blocajul amintit. Căci taskul expeditor aşteaptă pînă cînd taskul destinatar achită mesajul prin intermediul funcţiei "Reply Message ()". Ori, destinatarul deja nu mai este activ.



"Round Robin" este o ordonare circulară a listei "TCB"

Şi acest lucru ar fi uşor de evitat, dacă n-ar exista "amănunte" importante: "Yeld ()" nu se cheamă totdeauna explicit, ci este deseori apelată automat de funcţii deseori utilizate precum "Dialog Box ()" sau "Message Box". Deoarece nu poate să fie ceea ce nu are voie să fie, Windows oferă o funcţie numită "InSendMessage ()", cu care se poate constata dacă mai există mesaje neachitate. Deci dacă programatorul e cuminte (cu minte) şi testează totdeauna cu "InSendMessage ()" dacă apelul uneia din cele 8 funcţii critice (care apelează "Yield ()") este permis, atunci nu există blocaje. După cum se vede însă din practică, nu toţi programtorii respectă aceste indicaţii.

(I.F.)

Glosar

deadlock - două taskuri se aşteaptă reciproc, ducînd la blocajul întregului sistem

distribuitor - rutină sistem care asigură oprirea unui task şi lansarea altuia

eveniment - o acţiune externă, de care trebuie să se ţină seama (o tastare, o mişcare a mouse-ului, etc.)

FCFS - First Come, First Served (primul venit, primul servit)

HOL - Head of Line - toate taskurile sînt într-o coadă de aşteptare. Cel din capul listei e primul servit

non-preemtive - sistem multitasking fără control de timp sau priorităţi. Fiecare task trebuie să redea de bună voie controlul sistemului de operare, care poate lansa atunci un alt task

PCB - Process Control Block - formă extinsă a TCB

preemptive - sistem multitasking ghidat de timp sau de priorităţi. Un task dislocă un altul (preia procesul) cînd a trecut un anumit interval de timp sau pentru că are prioritate mai mare

prelucrare cvasi-paralelă - pe calculatoare cu un singur procesor, o prelucrare paralelă nu este posibilă. În schimb fiecare task este prelucrat pe o "felie de timp", după care se trece la următorul

prelucrare paralelă - mai multe taskuri sau sarcini parţiale se desfăşoară simultan pe mai multe procesoare, sub un sistem de operare unic.

round-robin - aranjament circular al listei de taskuri (după ultimul, urmează din nou primul)

scheduler - rutină sistem care decide, din mai multe taskuri posibile, care este cel selectat pentru continuare şi cît timp i se alocă.

sistem multiprocesor - calculator realizat cu mai mult de un procesor, în ultimul timp folosind mai ales transputere.

sistem multiuser - calculator ce poate fi folosit simultan de mai mulţi utilizatori, de la mai multe terminale (de exemplu Unix)

spooler - task ce se derulează în background (plan secund) care preia foarte repede în zona de proprie memorie un fişier ce urmează a fi tipărit, pentru a-l tipări apoi în întregime, consumînd pentru aceasta foarte puţin din timpul procesorului, disponibilizat astfel pentru alte activităţi în foreground (prim plan)

TCB - Task Control Block - zonă de date a unui task, în care sînt memorate toate datele interesante pentru sistemul de operare relative la task (nume, prioritate, stare, etc.)

Digital Research Multiuser-DOS 5.0

Reţelele sînt prea scumpe pentru multe întreprinderi. Se caută soluţii mai avantajoase - una din ele ar putea fi Multiuser - DOS 5.0 de la Digital Research, cu care pot lucra pînă la 8 terminale simultan.

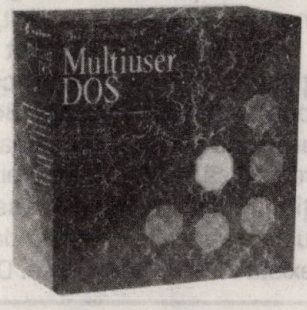
Din principiu, ceea ce un întreprinzător îşi doreşte cel mai mult la colaboratorii săi este o productivitate fără limite. De aceea, o dotare modernă a biroului, cu echipamente permanent disponibile, a devenit un imperativ. Cînd să se lege însă în reţea calculatoarele, mulţi întreprinzători nu îndrăznesc să facă investiţiile necesare. De aceea, deseori, în firmele mici există un singur calculator care - pînă acum - se porneşte doar de cîteva ori pe zi. Şi atunci, imediat, 3 colaboratori stau la rînd, pentru a-şi putea introduce datele. Dar timpul de aşteptare este timp pierdut.

Astfel că un sistem pe măsură ar putea arăta în felul următor: un calculator central - bine dotat, cu memorie multă (minim 2 Mbyte RAM, preferabil 4), disc de masă de capacitate mare (min 80 Mbyte) precum şi echipamente periferice - şi mai multe terminale "proaste", conectate ca staţii de lucru. Această "reţea" trebuie să fie uşor de instalat. Siguranţa în funcţionare este oricum o premisă indispensabilă.

Digital Research oferă o soluţie care, la prima privire, satisface toate aceste cerinţe: Multiuser - DOS 5.0 pentru circa 1500 DM. Rezistă la o verificare mai atentă?

Multiuser - DOS 5.0 este întîi de toate un sistem de operare MS - DOS compatibil. Foloseşte capacităţile procesorului 80386 - deci calculatorul central trebuie să fie dotat cu un cip 80386SX, 80386 sau 80486. Acest tip de procesor este capabil de multitasking, deci poate rezolva mai multe sarcini simultan. În modul de funcţionare multitasking, el emulează mai multe procesoare 8086, care rulează "simultan". Este exact ce foloseşte Multiuser - DOS. Atunci, pînă la 8 terminale "proaste" (staţii de lucru fără disc de masă şi UC) pot lucra deodată. Ele se leagă prin interfeţe seriale - deci calculatorul central trebuie să aibă suficiente racorduri.

Acest mod de transfer al datelor evidenţiază, aparent, un prim dezavantaj - viteza de lucru este mult redusă faţă de reţelele "adevărate". Şi totuşi, nu e chiar aşa. MU-DOS tratează fiecare PC conectat ca pe un terminal pur şi simplu, ignorînd complet "inteligenţa" disponibilă în acesta. Spre deosebire de multe



reţele locale, centralul nu oferă staţiei conectate date şi programe, spre a fi prelucrate pe procesorul staţiei, ci trimite pe linie numai "ecrane" gata pregătite, preluînd apoi de la terminal eventualele tastări. Centralul "cadoriseşte" staţiile conectate cu o parte din capacitatea sa de calcul şi preia integral execuţia tuturor programelor şi toate prelucrările de efectuat. Astfel, pe linie circulă un volum de date pentru care vitezele de transmisie realizabile cu o simplă legătură serială sînt absolut satisfăcătoare.

Funcţie de cablul serial folosit, se poate ajunge la ecrane "cu pureci", dacă există erori în transmisie (nu se face nici un fel de verificare, terminalele sînt considerate total neajutorate...) În mod normal, reducerea vitezei de transmisie rezolvă problema. Cabluri de calitate bună, scurte, pot lucra chiar pînă la 38.400 Baud, deşi 19.200 pare a fi o recomandare mai bună, chiar pentru cabluri de pînă la 10 m lungime. Sistemul "se vede" suficient de rapid chiar şi la 9600 Baud.

Este evident faptul că orice surplus de capacitate la terminale este risipă, dacă acestea urmează să fie folosite numai cu MU-DOS. XT - uri venerabile sau AT -uri descompletate sînt candidaţi ideali pentru acest scop.

Nu este posibilă nici integrarea suporţilor de date locali în reţea. Odată lansat "PCTERM", programul de emulare terminal, toate referinţele se adresează la periferice centralului. "A:" nu mai este unitatea de floppy cu care tocmai aţi lucrat, ci este cea de pe central - dacă vreţi ca datele de pe discheta Dvs. să ajungă la central, fie duceţi pur şi simplu discheta acolo, fie renunţaţi temporar la MU-DOS şi lansaţi un program de comunicaţie adecvat (FILELINK de ex., cunoscut deja din DR-DOS).

Sub controlul PCTERM, ecranul şi tastatura terminalului pot fi privite ca parte integrantă a centralului. Toate celelalte resurse locale (harddisk-uri, unităţi floppy, imprimantă, mouse etc.) sînt în hibernare. Chiar şi adaptorul grafic "doarme". Indiferent de rezoluţia de care este capabil adaptorul din terminal, centralul nu "prepară" în modul grafic, pentru terminalele conectate, ecrane care să depăşească rezoluţia unui CGA. Păcat, căci la ora actuală chiar şi cele mai ieftine PC-uri sînt dotate cu adaptoare mai bune.

Deşi în mod standard DOS lucrează numai cu maxim 4 interfeţe seriale, MU-DOS n-are probleme cu de 4 ori pe-atîtea. Drivere potrivite pentru 64 de cartele multiport (cartele ce diferă de interfaţa serială standard, avînd de exemplu posibilităţi de conectare pentru 8 sau 16 echipamente seriale) oferă

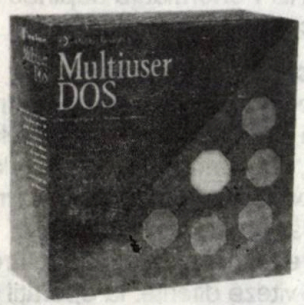
variate posibilităţi de realizare a legăturilor.

Pentru ca în sistem să nu poată pătrundă spioni, MU-DOS oferă un sistem de securizare administrat de utilizator. Se pot proteja, cu parole individuale, programe, fişiere de date, directoare întregi, hard-disk-uri sau dischete.

Există o comandă MAIL, prin care se pot transmite mesaje de la un terminal la altul. Dar rămîne în grija fiecărui utilizator să se "uite" în cutia poştală, să vadă dacă are ceva - sistemul nu-l anunţă în nici un fel.

CSPOOL poate administra mai multe imprimante deodată, pentru toate terminalele conectate - un serviciu indispensabil oricărei reţele.

Probleme cu utilizatori răuvoitori ar putea fi generate de comanda XSTOP. Aceasta permite oprirea



oricărei aplicaţii de pe orice terminal. Şi mai periculos este REBOOT - cu această comandă, se execută o restartare a centralului. Chiar şi atunci cînd comanda a fost dată de la un terminal. Conform manualului de utilizare, comanda ar trebui să fie la dispoziţia exclusivă a "super-utilizatorului" - cel de la central. Acest lucru se pare însă că nu este numaidecît adevărat. Dacă REBOOT nu se protejează printr-o parolă, atunci orice utilizator supărat poate termina toate activităţile în curs - brusc şi irevocabil. Rezultatul poate fi catastrofal pentru datele de pe sistem.

Spre deosebire de REBOOT, comanda SECURITY este într-adevăr la dispoziţia exclusivă a super-utilizatorului. Apelarea neautorizată a acesteia aduce pe ecran mesajul "only be used by the system administrator" - convingător (cînd nu eşti tu acela...)

Confortabil pe de altă parte este că fiecare din procesoarele 8086 virtuale este complet independent. Dacă - dintr-o tastare eronată de exemplu - un terminal "cade", restartarea se face, ca de obicei, tastînd <Ctrl> + <Alt> + <Del>. Ceilalţi utilizatori nu observă nimic din întreaga dandana (cu condiţia, totuşi, ca la instalare să se fi dezactivat opţiunea "Global Reset"-altfel.).

Ca şi pe central, orice terminal - chiar şi un XT prăpădit - poate comuta între mai multe programe prin simplă apăsare de taste, rezolvînd concomitent mai multe joburi deodată. Multitasking "real", de cea mai bună calitate. Funcţie de memoria disponibilă pe central, fiecare terminal poate rula deodată pînă la maxim 8 aplicaţii.

Sistemul de operare diferenţiază noţiunile de "station", "console" şi respectiv "session". O staţie este un terminal conectat la o interfaţă serială. Pentru fiecare staţie, pot fi deschise pînă la 8 sesiuni, fiecărei sesiuni corespunzîndu-i cîte o consolă. Fiecare sesiune constituie un domeniu DOS distinct, ce poate fi configurat individual, inclusiv memoria extinsă ce i se alocă. În afară de un AUTOEXEC.BAT local, aflat în directorul personal al utilizatorului, se pot face şi alte diferenţieri prin intermediul variabilelor de sistem, astfel încît la pornirea sistemului fiecare sesiune să fie pregătită pentru anumite sarcini.

Sesiunea care lucrează în "prim-plan" foloseşte în principiu întregul ecran - nu se lucrează, ca la Windows, în ferestre. Trecerea la alt ecran se face folosind o combinaţie de taste. Se poate tasta fie <Ctrl> + <Esc> , care aduce un meniu pe ecran, din care apoi se selectează aplicaţia dorită, fie <Alt> + <Esc>, care trece în sesiunea următoare, fie <Ctrl> + o cifră de pe blocul numeric al tastaturii, avînd ca efect trecerea directă în sesiunea cu numărul indicat.

O sesiune deschisă poate rula şi în "background" (în plan secund, în spate, fără să se vadă). Dacă nu este vorba de calcule multe, atunci existenţa ei nu impietează prea mult asupra performanţei generale. Există şi o "idle-detection" (detecţia unei stări în care programul nu face nimic util, aşteptînd de exemplu o tastă) care permite "liniştirea" unei sesiuni care risipeşte timpul procesorului. Opţiunea SUSPEND permite "îngheţarea" temporară a tuturor programelor ce nu lucrează în prim-plan. În acest caz, sigur că nu mai este vorba de multitasking.

Centralul acceptă moduri video diferite pentru sesiuni diferite, cît timp este vorba de moduri standard. Se poate lucra deci într-o sesiune cu un program grafic şi o rezoluţie VGA de 640 x 480 pixeli, în timp ce un altul lucrează în mod text iar un al treilea foloseşte modul CGA. Probleme apar la adaptoare VGA moderne, pe care MU-DOS nu le poate trata corect la trecerea dintr-o sesiune într-alta şi înapoi. Ecranul 'se murdăreşte' şi nu mai poate fi "curăţat" decît prin intermediul unor macrouri speciale.

Probleme reale şi deranjante există la lucrul cu mouse. Cît timp se foloseşte un mouse compatibil Microsoft şi se doreşte folosirea lui exclusiv pe central, totul e clar. Un singur mouse fizic, poate fi folosit, complet independent, în toate sesiunile, afectînd totdeauna numai sesiunea aflată în prim-plan.

Acest mouse poate fi disponibilizat şi altor utilizatori, care lucrează la terminale - fără însă a-l deconecta de la central. Ceea ce în practică înseamnă că toate terminalele care vor dori să lucreze cu aplicaţii care necesită mouse trebuie să se afle în imediata vecinătate a centralului, pentru a permite mutarea mouse-ului la terminalul ce are nevoie de el, fără a-l "scoate" din central. Sigur că se pot conecta mai multe mouse-uri la central - afectarea lor unui anumit terminal nu este o problemă şi se face prin procedura de instalare (setup).

(continuare în pag. 28)

Fileserver: alegere corectă

Dintre toate PC-urile dintr-o rețea, fileserver-ele sînt cele mai importante. Responsabile de păstrarea tuturor datelor, alegerii lor ar trebui să i se acorde o importanță deosebită. Nu numai viteza este importantă - un rol determinant îl mai au și lărgimea bus-ului, partea de rețea, carcasa și multe altele.

Server, în traducere directă, înseamnă servitor. Și fără server, nu se descurcă nici o rețea locală. Astfel, un fileserver este un pilon de bază în arhitectura unui LAN. Inițial, rolul său era de a păstra datele pentru toate stațiile de lucru reunite în rețea, servindule astfel pe acestea.

Și astăzi, acesta este unul din serviciile cele mai pretențioase pe care le oferă, chiar dacă acesta a cîștigat mult în complexitate (de exemplu datorită conceptului de client - server - computing). Importanța fileserverului nu a fost diminuată: la căderea server-ului, toată rețeaua stă; de aceea, la rețele mari se recomandă oglindirea discului de masă sau chiar dublarea fileserver-ului. Alegerii fileserverului ar trebui să i se acorde o importanță deosebită.

Fileserverul este "inima" întregii rețele, și pentru că pe el este instalat sistemul de operare al rețelei și toate programele și datele ce trebuie să fie disponibile la nivelul întregii rețele. În plus, foarte multe din caracteristicile importante pentru o rețea locală, cum ar fi securitatea și protecția datelor, depind fizic de tipul și dotarea fileserverului.

În fond, desigur, un fileserver nu este altceva decît un PC normal, deci cu UC, memorie operativă, disc de masă, etc. Și el poate fi apreciat după cîțiva factori de performanță, care, în legătură cu sarcinile sale funcționale, ar trebui să servească drept criterii de selecție. Acest lucru este la fel de important în rețele mici ca și în cele mari. Desigur cu diferența că în rețelele mari importanța fileserver-ului crește. Nu numai pentru că sînt conectate mai multe stații de lucru și se desfășoară mai multe aplicații, ci pentru că în astfel de cazuri se folosesc întotdeauna servere dedicate.

Adică: un PC este instalat ca server și aceasta este funcțiunea sa exclusivă, spre deosebire de server-ul nededicat, care simultan - ca activitate "secundară" - servește unui utilizator ca punct de lucru. Dacă în LAN-uri cu servere nededicate se definesc PC-uri servere, ele pot funcționa atît ca servere cît și ca puncte de lucru; de aceea, "căderea" unui server nu este așa de tragică ca în rețele mari, cu server dedicat.

Criterii de selecție

După cum am mai pomenit, fileserverele sînt în fond PC-uri normale, deși parțial special configurate pentru rețele, lucru asupra căruia vom reveni. Ceea ce face posibilă aprecierea lor după criterii aplicabile oricărui astfel de echipamnet.

Procesorul și frecvența de ceas: constituie desigur unul din cele mai importante criterii de selecție; lucru pe care îl realizează desigur oricine a făcut vreodată trecerea de la 8088 la 80286. Și cît de mult s-ar repercuta acest lucru asupra vitezei întregii rețele, ar trebui să fie evident. În timp ce 80286 lucrează atît intern cît și extern pe 16 biți, procesoarele 80386 respectiv 68030 de la Motorola sînt orientate pe 32 biți. Și astfel de calculatoare "state - of -the - art" (cele mai bune la momentul discuției) ar trebui să fie avute în vedere pentru cine ține la rețeaua lui. Dacă nu se face direct trecerea la procesoare de tip 486. Relativ la procesoare Intel sau Motorola: în domeniul PC-urilor, procesoarele Intel sînt în continuare regula. Cu ele, există garanția că nu numai cele mai importante sisteme de operare de pe PC-uri (DOS, OS/2) sînt funcționale, ci și sistemele de operare în rețea le sînt adaptate. Astfel, decizia pentru procesoarele Intel oferă certitudinea că întregul soft deja existent "merge" - desigur, în măsura în care e apt de funcționare în rețea.

Frecvența ceasului contribuie și ea la performanța unui PC. Oricum însă, performanța nu crește linear cu frecvența de ceas. Deci, în mod normal, un PC la 30 MHz nu e de două ori pe atît de iute ca unul la 15 MHz. Performanța depinde puternic de o altă componentă.

Cicli de wait (așteptare): un procesor care lucrează la o frecvență de ceas mare costă relativ puțin, în comparație cu celelalte componente electronice, în mod special memoriile. De aceea, deseori este nevoie de o adaptare la două viteze diferite: la operații pur interne, procesorul lucrează la toată viteza, deci execută o operație la fiecare ciclu de ceas, în timp ce la o operație externă (acces la memoria principală sau cartele suplimentare) trebuie să prelungească durata operației pe mai mulți cicli. De aici numele de ciclu wait. La alegerea unui server, nu ar trebui deci să se pună bază doar pe frecvența de ceas a procesorului. Un PC normal la 10 MHz fără ciclu de așteptare este de exemplu mai rapid decît unul la 12 MHz, dar care lucrează cu un ciclu de așteptare atunci cînd accedează memoria. "State - of - the - art" sînt fileservere ce lucrează la peste 30 MHz, unde trebuie să se țină seama însă și de cicli de așteptare.

Memoria principală: cu cît sînt mai mari rețelele, cu atît cantități mai mari de date trebuiesc prelu-

crate în memoria principală - şi memoria cache trebuie luată în considerare. Două puncte sînt importante. Mai întîi, care este memoria maximă adresabilă. Astfel de exemplu, la reţele mai mari pot fi necesare pînă la 16 Mbyte de memorie, pentru a garanta o prelucrare optimă a datelor.

În al doilea rînd, dacă implementarea acesteia se poate face direct pe cartela principală sau printr-un racord special, sau se face prin cartele suplimentare. Căci acestea din urmă implică, în mod normal, iarăşi, cicli de wait.

Bus-ul sistemului şi lărgimea bus-ului: la alegerea bus-ului sistemului, există între timp posibilitatea opţiunii între arhitectura (tradiţională) ISA, arhitectura MCA (microchannel) sau EISA (extended ISA). Din punct de vedere al ratei de transfer, nu se poate face o recomandare globală. Deoarece decizia odată luată determină posibilităţile ulterioare de extensie (cel puţin între ISA/EISA pe de o parte şi MCA pe de altă parte), alegerea ar trebui luată funcţie de strategia generală a întreprinderii relativă la achiziţionarea de PC-uri.

Lărgimea bus-ului a fost pomenită deja în legătură cu tipul procesorului. Ea are importanţă şi pentru că ea determină cantitatea de date care se vehiculează într-un ciclu de ceas între UC şi o cartelă de extensie. La un bus de 32 biţi, această cantitate e desigur semnificativ mai mare decît la unul de 16 sau 8 biţi. Pentru fileserver, ar trebui luate în considerare PC-uri care permit utilizarea unei cartele adaptor de 32 biţi. La punctele de lucru, desigur că acest lucru nu trebuie neapărat să se impună.

Harddisk-uri şi subsisteme de memorie de masă de capacitate mare: Pe lîngă harddisk-uri, tot mai mult se folosesc şi discuri optice. Deoarece datele efective cresc totdeauna rapid, un PC care permite folosirea numai a unui sau două harddisk-uri nu este indicat ca fileserver. Pentru a fi pregătit pentru viitor, ar trebui să existe loc pentru cel puţin opt.

Subsistemele disc oferă desigur şi opţiunea capacităţii de memorare liber extensibile - din punct de vedere al fileserver-ului însă, acest lucru prezintă inconveniente. E nevoie astfel de întreţinerea mai multor sisteme, e necesară o carcasă şi o alimentare proprie. Dacă fileserver-ul dispune de o sursă neîntreruptibilă - ceea ce este strict necesar dealtfel - aceasta nu prea e de mare folos la harddisk-urile externe.

Posibilităţile de extensie: într-un server trebuie să fie disponibile multe slot-uri libere. Aceasta nu numai din motive de extensie a sistemului (ca de exemplu eventuale cartele RAM, controllere suplimentare sau adaptoare serial-paralele etc.), ci şi pentru eventuala folosire a unor cartele de adaptare LAN. Astfel, se întîmplă deseori să apară necesitatea "legării" mai multor reţele - prin "bridge"-uri (poduri) de exemplu. Atunci, se poate întîmpla să trebuiască implantate pînă la 4 cartele adaptor într-un server.

Carcasa/design-ul: la fileserver, cu certitudine apare necesitatea schimbării, adăugirii sau eliminării unor părţi. De aceea, este important ca PC-ul utilizat ca server să fie uşor accesibil, preferabil din mai multe părţi. Deasemenea, toate componentele interne ar trebui să fie uşor şi simplu accesibile. Totodată ar trebui să fie oferite mecanisme de protecţie care să facă imposibil - sau măcar foarte dificil - accesul unui neavizat la atît de importantul server.

Se poate desigur - şi în anumite circumstanţe chiar ar trebui - instala server-ul într-o încăpere separată. Dacă acest lucru nu este posibil, carcasa ar trebui să poată fi măcar încuiată.

Tocmai la fileserver ar trebui avute în vedere şi alte măsuri care servesc siguranţei. Astfel, unitatea floppy ar putea fi integrată sub panoul frontal, care să poată fi încuiat. Trebuie să existe suficiente ventilatoare care să protejeze la supraîncălzire, căci fileserver-ul merge de obicei zi-noapte. Sînt avantajoase desigur şi afişaje cu LED-uri care să permită cuprinderea dintr-o singură privire a stării celor mai importanţi parametri de funcţionare.

IBM PS/2-Modell 95

Fiabilitatea: este desigur factorul cel mai important. Acest lucru începe cu sursa de alimentare şi se continuă cu robusteţea electronicii, proiectată pentru o utilizare non - stop. Cel tîrziu aici, se poate separa făina de tărîţe. În două privinţe.

În primul rînd, PC-urile "de marcă", cu care deja s-au făcut experienţe pozitive ca fileservere, se deosebesc de cele "no name", care sînt ieftine, dar deseori nu conţin o electronică bine acordată. Nume ce prezintă garanţii în acest domeniu sînt Compaq, NCR şi IBM. Chiar

Compaq Systempro

dacă sînt mai scumpe: economiile ar trebui făcute cu discernămînt.

În al doilea rînd, se deosebesc PC-urile de marcă, de PC-urile special concepute ca file-servere. Căci există producători care s-au specializat în producerea de fileservere. Corespunzător, multe din caracteristicile discutate mai sus se regăsesc în acestea. Ca nume, ar merita pomenite 3COM, NCI, Monolithics sau AST. Multe echipamente oferă pe lîngă procesoare foarte rapide şi posibilitatea de a utiliza mai multe UC-uri de mare performanţă (386 şi/sau 486) în mod mixt, într-un singur calculator, de exemplu Compaq cu Systempro sau Parallan cu Hyperserver. În măsura în care sistemele de operare "colaborează", se pot obţine astfel caracteristici care le ating pe cele ale calculatoarelor mari.

Dacă acestea nu sînt însă necesare - ceea ce, de regulă, este adevărat în marea majoritate a reţelelor locale, care numără la pînă în 10 puncte de lucru reunite - se recomandă folosirea unui PC cu 386 sau 486 ca fileserver, cu considerarea cerinţelor punctate mai sus. "Creşterea" unui LAN de regulă este preprogramată - astfel că un fileserver performant merită avut în vedere de la bun început.

(I.F.)

Digital Research Multiuser - DOS 5.0

(continuare din pag. 25)

Relativ la tema RAM, MU-DOS oferă mai multe posibilităţi de configurare optimală. Digital Research susţine că la o încărcare medie şi o configurare optimală, fiecărei sesiuni i se poate dedica 80% din capacitatea de calcul medie. Chiar dacă la nevoie "valori medii" este o noţiune la fel de extensibilă ca guma, diverse teste de performanţă lasă o impresie bună.

Sigur că sînt probleme principiale cu aplicaţii ca 'Windows 3.0' în modul extins sau 'UltraScript', care comută 80386 în Protected Mode şi depind foarte mult de memoria extinsă. Deoarece MU-DOS lucrează în mod virtual, conceput special pentru utilizarea multitasking a UC-urilor cu 80386, întreaga memorie este deja protejată şi nu mai poate fi trecută încă odată în această stare de alte programe.

Pentru instalarea pachetelor de programe cele mai des folosite, există indicaţii în manualul de utilizare (pentru Paradox 3.5, Word 5.0, Lotus 1-2-3 2.2 (Server Edition), Dataease 4.2 (LAN), Quattro-Pro 2.0, Lotus Manuscript 2.1 (LAN), WordPerfect 5.1, Letter-Perfect 1.0, Q&A 3.0 (LAN), SuperCalc 5, Framework III 1.1 şi dBase IV 1.1).

Digital Research îşi protejează sistemul - comercializat la 1500 DM - cu un "dongle", care trebuie introdus în prima interfaţă paralelă. Ca la multe alte astfel de scheme de protecţie, imprimanta legată la dongle trebuie să fie neapărat pe automat (online) pentru ca sistemul, la pornire, să depăşească faza de verificare a protecţiei. E discutabilă această soluţie - imaginaţi-vă ce-ar fi dacă oricare aplicaţie mai bună ar fi protejată cu un astfel de dispozitiv. S-ar deschide o nouă piaţă pentru produse de stabilizare a conectorilor disponibili pe panoul din spate al PC - urilor ...

MU-DOS poate lucra cu harddisk-uri de pînă la 2 Gbyte (!). Ceea ce înseamnă peste 200 de discuri de 100 Mbyte... Pentru unităţi CD-ROM uzuale sînt disponibile drivere adecvate, astfel încît merită să se investească în central, făcînd din el o veritabilă centrală de date.

Per ansamblu, Multiuser - DOS 5.0 de la Digital Research se dovedeşte a fi o soluţie performantă, cu un multitasking aproape perfect. Sistem multiutilizator, permiţînd instalarea unor locuri de muncă ieftine pentru pînă la 8 colaboratori, el se pretează îndeosebi la eficientizarea unor laptop-uri mai slabe sau a unor XT-uri vechi, care deseori dispun numai de adaptoare video CGA sau monocrome şi au doar o unitate floppy în configuraţie.

Programatorii de la Digital Research au realizat un sistem "curat", care în mod cert va avea un impact serios asupra pieţii. Totuşi, ar trebui să rezolve neapărat, şi nu prea tîrziu, şi lucrul cu cartele grafice de rezoluţie înaltă, care sînt, în momentul de faţă, standarde acceptate. Altfel, la prima ocazie, din nou Microsoft va culege toţi laurii. Cum s-a mai întîmplat, de altfel.

(I.F.)

OLIVETTI * SZEGED

CAUTĂ
PARTENER DE AFACERI ÎN
VEDEREA UNEI COLABORĂRI
COMUNE ÎN CONDIŢII
DEOSEBIT DE AVANTAJOASE!

OFERĂ:

ÎNTREAGA GAMĂ
DE ECHIPAMENT DE CALCUL,
COPIERE ŞI BIROTICĂ
(COMPUTERE, COPIATOARE,
MAŞINI DE CALCULAT,
DESK-TOP PUBLISHING,
TELEFAX-URI,
MAŞINI DE SCRIS, ETC.)

INNOV TECHNIK Ltd.
6722 SZEGED, Gutenberg u. 3
HUNGARY
Tel/Fax: (36)62-19026

Lista comenzilor dBase IV

! - identic cu RUN
; - următoarea linie sursă este privită ca o linie de continuare a liniei curente
? - afişează conţinutul unor cîmpuri sau expresii şi face un avans de linie
?? - afişează conţinutul unor cîmpuri sau expresii dar nu face şi un avans de linie
??? - trimite comenzi de imprimare direct la imprimantă
@..CLEAR - şterge zone ale ecranului
@..FILL - modifică atributele de culoare în domeniul specificat de pe monitor
@..SAY..GET - afişează sau preia conţinutul unei variabile sau al unui cîmp
@..TO - desenează o linie sau un cadru între coordonatele specificate
***** - ca prim caracter într-o linie marchează linia respectivă ca pe o linie de comentariu
& - caracter macrou, determină înlocuirea numelui unei variabile cu conţinutul ei
&& - marchează restul liniei ca şi comentariu
ACCEPT - cere o valoare pentru o variabilă caracter afişată pe monitor
ACTIVATE MENU - activează un meniu linie
ACTIVATE POPUP - activează un meniu pop-up
ACTIVATE SCREEN - activează întregul domeniu monitor
ACTIVATE WINDOW - activează o fereastră
APPEND - inserează noi articole în fişierul curent
APPEND FROM - preia articole dintr-un alt fişier în fişierul curent
APPEND FROM ARRAY - adaugă articolele dintr-un tablou (array) la fişierul curent
APPEND MEMO - preia un fişier de text într-un cîmp memo
ASSIST - apelează centrul de regie
AVERAGE - calculează media aritmetică
BEGIN TRANSACTION - porneşte o tranzacţie
BROWSE - afişează datele sub formă de tabelă
CALCULATE - efectuează anumite calcule într-un fişier
CALL - apelează un modul extern
CANCEL - întrerupe un fişier de comenzi
CASE - parte a construcţiei DO CASE
CHANGE - identic cu EDIT
CLEAR - şterge monitorul sau parametrul specificat
CLOSE - închide fişiere
COMPILE - compilează un fişier
CONTINUE - caută următoarea noţiune căutată cu LOCATE
CONVERT - converteşte un fişier DBF pentru a fi utilizat în reţea
COPY - copiază un fişier, sau părţi dintr-un fişier, într-alt fişier
COPY FILE - copiază un fişier
COPY INDEXES - copiază fişiere NDX într-un fişier MDX
COPY MEMO - copiază un cîmp memo într-un fişier text
COPY STRUCTURE - copiază structura unui fişier DBF
COPY STRUCTURE EXTENDED - copiază o formă specială a structurii unui fişier DBF
COPY TAG - converteşte intrările dintr-un fişier MDX în fişiere NDX
COPY TO ARRAY - copiază date dintr-un fişier DBF într-un tablou
COUNT - contorizează articolele unui fişier DBF care satisfac o condiţie dată
CREATE APPLICATION - apelează generatorul de programe pentru crearea unei aplicaţii
CREATE FROM - creează un nou fişier DBF dintr-o structură de date creată cu COPY STRUCTURE EXTENDED
CREATE LABEL - apelează generatorul de etichete
CREATE QUERY - identic cu CREATE VIEW
CREATE REPORT - apelează generatorul de rapoarte
CREATE SCREEN - apelează generatorul de machete
CREATE STRUCTURE - apelează programul de definire a unui fişier DBF
CREATE VIEW - apelează programul pentru definirea unui fişier VIEW
CREATE VIEW FROM ENVIRONMENT - memorează mediul curent de lucru ca fişier VIEW
DBLINK - programul extern cuprinde mai multe fişiere obiect
DEACTIVATE MENU - dezactivează un meniu linie
DEACTIVATE POPUP - dezactivează un meniu pop-up
DEACTIVATE WINDOW - dezactivează o fereastră
DEBUG - apelează depanatorul (debugger)
DECLARE - defineşte un tablou
DEFINE BAR - defineşte o opţiune pentru un meniu pop-up
DEFINE BOX - defineşte un chenar pentru o expresie
DEFINE MENU - defineşte un meniu linie
DEFINE PAD - defineşte o opţiune pentru un meniu linie
DEFINE POPUP - defineşte un meniu pop-up
DEFINE WINDOW - defineşte o fereastră
DELETE - poziţionează indicatorul de ştergere pentru articole
DELETE FILE - şterge un fişier
DELETE TAG - şterge o intrare într-un fişier MDX
DIR - afişează conţinutul unui director
DIRECTORY - identic cu DIR
DISPLAY - corespunde cu LIST dar se opreşte după fiecare pagină afişată
DISPLAY FILES - corespunde cu LIST FILES dar se opreşte după fiecare pagină afişată
DISPLAY HISTORY - corespunde cu LIST HISTORY dar se opreşte după fiecare pagină afişată
DISPLAY MEMORY - corespunde cu DISPLAY MEMORY dar se opreşte după fiecare pagină afişată
DISPLAY STATUS - corespunde cu LIST STATUS dar se opreşte după fiecare pagină afişată
DISPLAY STRUCTURE - corespunde cu LIST STRUCTURE dar se opreşte după fiecare pagină afişată
DISPLAY USERS - corespunde cu LIST USERS dar se opreşte după fiecare pagină afişată
DO - apelează un program sau o procedură
DO CASE - deschide un bloc 'case'
DO WHILE - porneşte un ciclu "atît timp cît"
EDIT - comută în modul editare
EJECT - determină un salt de pagină la imprimantă
EJECT PAGE - determină un salt de pagină controlat la imprimantă
ELSE - parte a construcţiei IF
END TRANSACTION - sfîrşitul buclei BEGIN TRANSACTION
ENDCASE - parte a construcţiei DO CASE
ENDDO - parte a construcţiei DO WHILE
ENDIF - parte a construcţiei IF
ENDPRINTJOB - sfîrşitul buclei PRINTJOB
ENDSCAN - sfîrşitul buclei SCAN
ENDTEXT - sfîrşitul buclei TEXT
ERASE - şterge un fişier
EXIT - determină ieşirea dintr-o buclă DO WHILE sau SCAN
EXPORT - converteşte un fişier DBF într-un format străin
FIND - caută o noţiune într-un fişier index
FUNCTION - denumeşte o funcţie autodefinită
GET - vezi: @..SAY..GET
GO - salt la un anumit articol
GOTO - vezi: GO
HELP - apelează sistemul de mesaje de ajutor
IF - deschide un bloc "dacă"
IMPORT - converteşte un format străin în format DBF

INDEX - crează un fişier index
INPUT - cere introducerea de la tastatură a valorii unei variabile numerice pe monitor
INSERT - inserează un articol vid în fişierul curent
JOIN - crează un nou fişier DBF din două fişiere DBF
KEYBOARD - pune caractere în buffer-ul de tastatură
LABEL FORM - porneşte o imprimare de etichetă
LIST - afişează articole, fără întrerupere
LIST FILES - afişează fişiere, fără întrerupere
LIST HISTORY - afişează ultimele comenzi tastate, fără întrerupere
LIST MEMORY - listează variabilele de memorie, fără întrerupere
LIST STATUS - listează starea curentă, fără întrerupere
LIST STRUCTURE - afişează structura fişierului curent, fără întrerupere
LIST USERS - afişează utilizatorii din reţea, fără întrerupere
LOAD - încarcă un modul extern
LOCATE - caută o noţiune într-un fişier DBF
LOGOUT - anunţă ieşirea unui utilizator din reţea
LOOP - parte a construcţiei DO WHILE
MODIFY APPLICATION - corespunde cu CREATE APPLICATION dar serveşte doar la modificări
MODIFY COMMAND - apelează editorul de programe
MODIFY FILE - corespunde cu MODIFY COMMAND, dar serveşte numai la modificări
MODIFY LABEL - corespunde cu CREATE LABEL, dar serveşte numai la modificări
MODIFY QUERY - corespunde cu CREATE QUERY, dar serveşte numai la modificări
MODIFY REPORT - corespunde cu CREATE REPORT, dar serveşte numai la modificări
MODIFY SCREEN - corespunde cu CREATE SCREEN, dar serveşte numai la modificări
MODIFY STRUCTURE - corespunde cu CREATE STRUCTURE, dar serveşte numai la modificări
MODIFY VIEW - corespunde cu CREATE VIEW, dar serveşte numai la modificări
MOVE WINDOW - mută o fereastră pe ecran
NOTE - marchează o linie ca linie de comentariu
ON ERROR - ramificaţie spre o rutină de tratare a erorilor în cazul apariţiei unei erori dBase IV
ON ESCAPE - după apăsarea tastei <Esc> programul execută secvenţa de comenzi care urmează după ON ESCAPE
ON KEY - după apăsarea tastei desemnate se execută secvenţa de comenzi care urmează
ON PAD - apelează un meniu pop-up dintr-un meniu linie
ON PAGE - execută anumite comenzi la imprimarea anumitor pagini
ON READERROR - în cazul unor erori de intrare execută comanda dată
ON SELECTION PAD - tratează o intrare dintr-un meniu linie
ON SELECTION POPUP - tratează o intrare dintr-un meniu pop-up
OTHERWISE - parte a construcţiei DO CASE
PACK - şterge dintr-un fişier toate articolele marcate ca şterse
PARAMETERS - atribuie unor variabile valorile parametrilor de apel din programul apelant
PLAY MACRO - execută un macro specificat
PRINTJOB - porneşte o listare la imprimantă
PRIVATE - limitează valabilitatea variabilelor la procedura sau funcţia curentă
PROCEDURE - defineşte o procedură autodefinită
PROTECT - apelează sistemul de protecţie dBase
PUBLIC - defineşte variabilele valabile în întregul program
QUIT - termină sesiunea dBase
READ - citeşte toate variabilele enumerate după GET
RECALL - îndepărtează marcajele de articole şterse
REINDEX - crează din nou indexul activ
RELEASE - şterge variabile din memorie
RELEASE MENUS - şterge meniurile linie
RELEASE POPUPS - şterge meniurile pop-up
RELEASE SCREENS - şterge ecranele memorate
RELEASE WINDOWS - şterge ferestre
RENAME - modifică numele unui fişier
REPLACE - înlocuieşte valoarea unor cîmpuri cu noi valori
REPLACE FROM ARRAY - preia înregistrările dintr-un tablou în fişierul curent
REPORT FORM - porneşte listarea unui raport
RESET - elimină marcajele care au fost poziţionate cu o comandă de tranzacţie şi care nu au fost îndepărtate încă
RESTORE - încarcă variabilele memorat
RESTORE MACROS - încarcă macrourile memorate
RESTORE SCREEN - încarcă un ecran memorat
RESTORE WINDOW - încarcă ferestrele memorate
RESUME - relansează un program întrerupt
RETRY - relansează o comandă care nu a fost executată
RETURN - revenire în programul apelant
ROLLBACK - derulează înapoi acţiunile unei tranzacţii
RUN - apelează o comandă a sistemului de operare
SAVE - memorează variabilele din memorie ca fişier
SAVE MACROS - memorează macrourile ca fişier
SAVE SCREEN - memorează conţinutul ecranului curent
SAVE WINDOW - memorează o fereastră ca fişier
SAY - vezi: @..SAY
SCAN - prelucrează articole care satisfac o anumită condiţie
SEEK - caută o noţiune într-un fişier index
SELECT - selectează un domeniu de lucru
SET - apelează un meniu pentru specificarea paramerilor de setare a mediului de lucru
SHOW MENU - afişează un meniu linie
SHOW POPUP - afişează un meniu pop-up
SKIP - mută pointerul articol în fişierul curent
SORT - sortează articolele într-un nou fişier
STORE - stochează o valoare într-o variabilă
SUM - adună numere într-un nou fişier
SUSPEND - întrerupe un program dBase
TEXT - marchează începutul unui bloc de text
TOTAL - adună valorile numerice ale unui fişier într-un al doilea fişier
TYPE - afişează conţinutul unui fişier text
UNLOCK - deblochează articole şi fişiere în reţea
UPDATE - înlocuieşte conţinutul unor cîmpuri cu valori din alte fişiere DBF
USE - deschide un fişier DBF
WAIT - aşteaptă apăsarea unei taste
WHILE - parte a construcţiei DO WHILE
ZAP - şterge toate articolele dintr-o bază de date

Lista operatorilor dBase IV

+	- adunare	>= sau =>	- mai mare sau egal
-	- scădere	<= sau =<	- mai mic sau egal
*	- înmulţire	$	- conţinut în
/	- împărţire	.and.	- şi
** sau ^	- ridicare la putere	.or.	- sau
<	- mai mic	.not.	- nu
>	- mai mare		
=	- egal		
<>	- diferit		

Lista funcţiilor dBase IV

ABS() - returnează valoarea absolută a unui număr
ACCESS() - determină drepturile de acces ale utilizatorului
ACOS() - calculează arccosinusul unui unghi
ALIAS() - returnează numele alias al unui domeniu de lucru
ASC() - returnează codul ASCII al unui caracter
ASIN() - calculează arcsinusul unui unghi
AT() - dă poziţia de început a unui şir de caractere în cadrul unui alt şir de caractere
ATAN() - calculează arctangenta unui unghi
ATN2() - calculează arctangenta unui unghi
BAR() - dă numărul opţiunii pop-up selectate
BOF() - indică începutul unui fişier
CALL() - apelează un modul extern
CDOW() - returnează ziua din săptămînă a unei date
CEILING() - rotunjeşte un număr la valoarea întreagă imediat superioară
CERROR() - dă codul de eroare al ultimei erori de compilare
CHANGE() - determină orice modificare a articolului curent de către un alt utilizator în reţea
CHR() - returnează caracterul ASCII indicat
CMONTH() - returnează numele lunii unei date
COL() - returnează numărul coloanei poziţiei curente a cursorului
COMPLETED() - specifică dacă o tranzacţie a fost încheiată cu succes
COS() - calculează cosinusul unui unghi
CTOD() - converteşte o succesiune de caractere într-o dată
DATE() - returnează data sistem
DAY() - returnează ziua unei date
DBF() - returnează numele DBF al unui fişier deschis
DELETED() - determină dacă un articol a fost şters
DIFFERENCE() - returnează gradul de corespondenţă a două şiruri
DISKSPACE() - returnează spaţiul liber de pe unitatea de disc curentă
DMY() - converteşte o dată într-o succesiune de caractere
DOW() - returnează ziua din săptămînă a unei date
DTOC() - converteşte o dată într-o succesiune de caractere
DTOR() - converteşte mărimea în grade a unui unghi în radiani
DTOS() - converteşte o dată într-o variabilă caracter pentru indexare
EOF() - indică sfîrşitul unui fişier
ERROR() - returnează codul de eroare al ultimei erori semnalate
EXP() - calculează e la puterea unui număr
FIELD() - dă numele cîmpului dorit dintr-un fişier
FILE() - determină existenţa fişierelor indicate
FIXED() - converteşte numerele de tip G în numere de tip N
FKLABEL() - dă numele tastei funcţionale dorite
FKMAX() - dă numărul tastelor funcţionale programabile
FLOAT() - converteşte numere de tip N în numere de tip G
FLOCK() - blochează un fişier într-o reţea
FLOOR() - rotunjeşte un număr la întregul imediat inferior
FOUND() - specifică dacă o căutare s-a încheiat cu succes
FV() - calculează valoarea viitoare a unor plăţi, inclusiv dobînzile, într-un anumit interval de timp
GETENV() - returnează caracteristicile sistem dorite
IIF() - reacţionează la o condiţie
INKEY() - returnează numărul tastei apăsate
INT() - returnează partea întreagă a unui număr
ISALPHA() - determină faptul dacă primul caracter dintr-un şir este o literă
ISCOLOR() - determină faptul dacă în calculator este instalat un adaptor grafic color
ISLOWER() - determină faptul dacă primul caracter a unui şir este o literă mică
ISMARKED() - indică dacă un fişier este implicat într-o tranzacţie
ISUPPER() - determină faptul dacă primul caracter al unui şir este o literă mare
KEY() - dă expresia de cheie a unui fişier index
LASTKEY() - returnează codul ultimei taste apăsate
LEFT() - returnează partea stîngă a unui şir de caractere
LEN() - dă lungimea unui cîmp sau a unei variabile
LIKE() - indică dacă un şir de caractere este conţinut într-un alt şir de caractere
LINENO() - dă numărul de linie dintr-un program în care a apărut o eroare
LKSYS() - dă numele utilizatorului unui fişier într-o reţea
LOCK() - identic cu RLOCK()
LOG() - calculează logaritmul în baza e al unui număr
LOG10() - calculează logaritmul în baza 10 al unui număr
LOOKUP() - caută o noţiune într-un alt fişier
LOWER() - converteşte toate literele dintr-un şir în litere mici
LTRIM() - elimină spaţiile de început dintr-un şir
LUPDATE() - returnează data ultimei modificări a unui fişier DBF
MAX() - determină maximul dintre două expresii numerice
MDX() - returnează numele fişierelor MDX deschise
MDY() - converteşte o dată într-un şir de caractere
MEMLINES() - dă numărul de linii pentru un cîmp memo
MEMORY() - returnează mărimea memoriei rămasă liberă
MENU() - returnează numele meniului linie activ
MESSAGE() - returnează mesajul de eroare pentru o eroare apărută
MIN() - determină minimul dintre două expresii numerice
MLINE() - returnează o linie a unui cîmp memo
MOD() - returnează restul unei împărţiri întregi
MONTH() - returnează luna unei date
NDX() - returnează numele fişierelor NDX deschise
NETWORK() - determină faptul dacă dBase IV lucrează în reţea
ORDER() - dă indexul activ
OS() - returnează versiunea sistemului de operare
PAD() - dă numele opţiunii selectate dintr-un meniu pop-up
PAYMENT() - calculează valoarea de plătit pentru un împrumut cu o dobîndă fixă pentru un număr dat de rate
PCOL() - returnează coloana poziţiei curente a imprimantei
PI() - dă valoarea pentru π
POPUP() - dă numele meniului pop-up activ
PRINTSTATUS() - specifică dacă imprimanta este pregătită pentru listare
PROGRAM() - specifică numele programului care este în curs de execuţie
PROMPT() - returnează textul punctului de meniu pop-up selectat
PROW() - returnează linia curentă a imprimantei
PV() - calculează valoarea actuală a unor plăţi cuprinzînd dobînzile pentru un interval de timp specificat
RAND() - generează un număr aleator
READKEY() - specifică tasta cu care a fost încheiată o comandă orientată ecran
RECCOUNT() - returnează numărul de articole dintr-un fişier DBF
RECNO() - returnează numărul de articol al articolului curent
RECSIZE() - returnează dimensiunea unui articol dintr-un fişier DBF
REPLICATE() - repetă o succesiune de caractere de mai multe ori
RIGHT() - returnează partea dreaptă a unui şir de caractere
RLOCK() - blochează un articol în reţea
ROLLBACK() - specifică dacă un rollback a fost încheiat cu succes
ROUND() - rotunjeşte un număr în sus sau în jos
ROW() - returnează numărul de linie al poziţiei cursor curente
RTOD() - converteşte valoarea unui unghi dat în radiani în grade
RTRIM() - elimină spaţiile de la sfîrşitul unui şir
SEEK() - caută o noţiune după un index
SELECT() - specifică numărul celui mai mare domeniu de lucru liber
SET() - returnează poziţionarea unei comenzi SET
SIGN() - returnează semnul unui număr
SIN() - calculează sinusul unui unghi
SOUNDEX() - returnează codul fonetic al unui şir de caractere
SPACE() - produce un şir de spaţii
SQRT() - calculează rădăcina pătrată dintr-un număr
STR() - converteşte valoarea unui număr într-un şir de caractere
STUFF() - înlocuieşte o parte a unui şir de caractere printr-o alta
SUBSTR() - returnează o parte a unui şir de caractere
TAG() - returnează numele indexului dorit (fişier TAG sau NDX)
TAN() - calculează tangenta unui unghi
TIME() - returnează ora sistem
TRANSFORM() - reprezintă variabilele prin intermediul unei măşti
TRIM() - elimină spaţiile de la sfîrşitul unui şir
TYPE() - specifică tipul unei variabile
UPPER() - converteşte toate literele unui şir în litere mari
USER() - returnează numele unui utilizator din reţea
VAL() - converteşte un şir de caractere într-un număr
VARREAD() - returnează numele variabilei sau cîmpului în curs de prelucrare
VERSION() - returnează versiunea dBase curentă
YEAR() - returnează anul unei date

Lista variabilelor sistem în dBase IV

_alignment - comutator pentru alinierea textelor
_box - stabileşte dacă vor fi sau nu imprimate chenarele definite cu DEFINE BOX
_indent - stabileşte indentarea primei linii a unui paragraf
_lmargin - stabileşte marginea din stînga pentru afişarea cu ?/??
_padvance - stabileşte modul de avans al hîrtiei
_pageno - stabileşte numărul paginii curente
_pbpage - stabileşte prima pagină de imprimat
_pcolno - stabileşte coloana ieşirii
_pcopies - stabileşte în cîte exemplare se va face imprimarea
_pdriver - stabileşte driver-ul de imprimantă care va fi folosit
_pecode - trimite coduri la imprimantă la sfîrşitul imprimării
_peject - controlează modul de salt de pagină în timpul unei imprimări cu PRINTJOB
_pepage - controlează numărul paginilor de imprimat
_pform - ataşează un fişier cu un format de imprimare
_plength - stabileşte numărul de linii pe pagină
_plineno - stabileşte linia curentă la ieşire
_ploffset - stabileşte marginea din stînga la imprimare
_ppitch - controlează distanţa dintre caractere
_pquality - comutator pentru calitatea imprimării
_pscode - trimite coduri la imprimantă înainte de începerea listării
_pspacing - stabileşte distanţa dintre linii
_pwait - stabileşte dacă imprimanta va aştepta după fiecare pagină listată
_rmargin - stabileşte marginea din dreapta la afişarea cu ?/??
_tabs - poziţionează tabulatorii
_wrap - stabileşte dacă se face "ruperea" liniilor lungi

Lista comenzilor SET în dBase IV

SET ALTERNATE - copiază ieşirea pe ecran şi într-un fişier
SET AUTOSAVE - stabileşte dacă modificările de articole să fie efectuate imediat şi pe disc sau nu
SET BELL - stabileşte durata şi frecvenţa alarmei sonore, putînd şi să o anuleze
SET BLOCKSIZE - stabileşte dimensiunea blocurilor cîmpurilor memo şi fişierelor MDX
SET BORDER - stabileşte forma chenarului ferestrelor şi meniurilor
SET CARRY - stabileşte dacă, şi care, conţinuturi de cîmpuri vor fi preluate din articolul precedent într-un nou articol
SET CATALOG - deschide un fişier catalog
SET CENTURY - stabileşte forma de afişare a anilor (cu sau fără sute)
SET CLOCK - controlează afişarea orei sistem în programe
SET COLOR - stabileşte culorile de afişare pe monitor
SET CONFIRM - stabileşte dacă cursorul va sări sau nu în cîmpul următor la terminarea unui cîmp
SET CONSOLE - stabileşte dacă se face sau nu afişarea pe monitor
SET CURRENCY - stabileşte simbolul monezii utilizate
SET CURSOR - stabileşte dacă cursorul va fi afişat sau nu
SET DATE - stabileşte formatul datei
SET DEBUG - redirecţionează afişările comenzii SET ECHO la imprimantă
SET DECIMALS - stabileşte numărul poziţiilor zecimale afişate
SET DEFAULT - stabileşte unitatea de disc curentă
SET DELETED - stabileşte dacă să fie ignorate sau nu şi articolele şterse
SET DELIMITERS - stabileşte delimitatorii pentru afişarea cîmpurilor
SET DESIGN - stabileşte dacă generatorul de programe poate fi lansat din centrul de regie
SET DEVELOPMENT - activează validarea datelor de creere a unor fişiere program originale şi compilate
SET DEVICE - stabileşte perifericul de ieşire pentru comanda @..SAY
SET DIRECTORY TO - stabileşte unitatea şi directorul curent
SET DISPLAY - stabileşte modul de afişare pe ecran
SET DOHISTORY - fără funcţie
SET ECHO - afişează pe ecran liniile de program executate
SET ENCRYPTION - activează cifrarea fişierelor
SET ESCAPE - stabileşte dacă tasta <Esc> este activă
SET EXACT - cere o corespondenţă exactă a două şiruri de caractere
SET EXCLUSIVE - deschide fişiere în reţea exclusiv
SET FIELDS - limitează cîmpurile de prelucrat
SET FILTER - limitează fişierul la articole care să respecte o anumită condiţie
SET FIXED - fără funcţie
SET FORMAT - atribuie o mască comenzilor orientate ecran
SET FULLPATH - activează afişarea integrală a căii (path)
SET FUNCTION - încarcă o succesiune de caractere pe o tastă funcţională
SET HEADING - activează afişarea antetelor de coloană
SET HELP - activează afişarea automată a mesajelor de help
SET HISTORY - activează şi stabileşte dimensiunea memoriei de comenzi
SET HOURS - stabileşte modul de afişare al orei sistem
SET INDEX - stabileşte indexul activ
SET INSTRUCT - activează afişarea de indicaţii
SET INTENSITY - activează reprezentarea evidenţiată a cîmpurilor în timpul introducerii
SET LOCK - stabileşte dacă articolele vor fi blocate automat în timpul prelucrării în reţea
SET MARGIN - stabileşte marginea din stînga la imprimare
SET MARK - stabileşte separatorul între ziua, luna şi anul unei date
SET MEMOWIDTH - stabileşte lăţimea de ieşire pentru un cîmp memo
SET MENU - fără funcţie
SET MESSAGE - afişează un mesaj al programului
SET NEAR - stabileşte dacă la o căutare a unei valori este necesară o potrivire exactă sau este suficientă asemănarea
SET ODOMETER - stabileşte intervalul contorului de articole
SET ORDER - stabileşte indexul curent
SET PATH - stabileşte calea (path-ul)
SET PAUSE - stabileşte afişarea datelor în modul SQL
SET POINT - stabileşte forma semnului zecimal
SET PRECISION - stabileşte precizia de calcul în dBase
SET PRINTER - stabileşte ieşirea pentru imprimare
SET PROCEDURE - activează un fişier de proceduri
SET REFRESH - stabileşte intervalul de timp după care va fi reîmprospătată imaginea afişată pe ecran cu datele actualizate la lucrul în reţea
SET RELATION - stabileşte o legătură între două fişiere
SET REPROCESS - stabileşte de cîte ori va fi repetată încercarea de executare a unei comenzi la lucrul în reţea în cazul în care prima încercare eşuează
SET SAFETY - împiedică suprascrierea din greşeală a fişierelor
SET SCOREBOARD - controlează afişarea setărilor pentru tastatură
SET SEPARATOR - stabileşte forma de reprezentare a miilor (separatorul)
SET SKIP - stabileşte în care fişier va fi mişcat pointerul articol în cazul fişierelor legate prin SET RELATION
SET SPACE - inserează un spaţiu între cîmpurile afişate
SET SQL - activează modul SQL
SET STATUS - activează afişarea liniei de stare în partea de jos a ecranului
SET STEP - execută un program pas cu pas
SET TALK - controlează afişarea mesajelor
SET TITLE - editează liniile de comentariu ale unui catalog
SET TRAP - apelează debugger-ul în cazul apariţiei unei erori în program
SET TYPEAHEAD - stabileşte mărimea buffer-ului de intrare
SET UNIQUE - stabileşte dacă un index va conţine toate cheile sau numai cheile unice
SET VIEW - deschide un fişier VIEW
SET WINDOW - ataşează o fereastră unui cîmp memo

Lista comenzilor SQL disponibile în dBase IV

ALTER TABLE - inserează coloane într-o tabelă
CLOSE - încheie accesul dBase la o tabelă
CREATE DATABASE - crează o nouă bază de date
CREATE INDEX - crează un index pentru una sau mai multe coloane sau pentru un view
CREATE SYNONYM - crează un sinonim pentru o tabelă sau pentru un view
CREATE TABLE - crează o nouă tabelă
CREATE VIEW - crează o tabelă virtuală (view)
DBCHECK - verifică corespondenţa dintre o tabelă SQL şi un fişier dBase IV
DBDEFINE - converteşte o bază de date dBase într-o tabelă a bazei de date SQL active
DECLARE CURSOR - defineşte un acces dBase la o tabelă
DELETE - şterge linii dintr-o tabelă sau dintr-un view
DROP DATABASE - şterge o bază de date
DROP INDEX - şterge un index
DROP SYNONYM - şterge un sinonim
DROP TABLE - şterge o tabelă
DROP VIEW - şterge un view
FETCH - sare la un acces dBase la o tabelă la intrarea următoare
GRANT - stabileşte drepturile de acces pentru o tabelă
INSERT - inserează linii într-o tabelă sau într-un view
LOAD DATA - importă date într-o tabelă SQL
OPEN - începe accesul dBase la o tabelă
REVOKE - şterge (revocă) drepturile de acces pentru o tabelă
ROLLBACK - derulează înapoi modificările unei tranzacţii neîncheiate
RUNSTATS - actualizează intrările statistice ale tabelei actuale
SELECT - lansează o interogare SQL
SHOW DATABASE - afişează bazele de date existente
START DATABASE - deschide o bază de date
STOP DATABASE - închide baza de date activă
UNLOAD DATA - exportă date dintr-o tabelă SQL
UPDATE - actualizează datele unei selecţii

Planificarea este alfa şi omega

OOP - Partea a cincea

În partea finală a acestui curs sînt examinate problemele care apar la planificarea ierarhiilor de obiecte.

Primele trei părţi ale acestui curs s-au ocupat cu bazele programării orientate obiect. În partea a patra au fost prezentate tehnicile de construire a unei ierarhii. În partea finală a cursului va fi demonstrat modul de cuplare al unor noi obiecte într-o ierarhie.

În partea a patra a cursului neam concentrat atenţia asupra modului de realizare a Unit-ului **BS_Tools**. Ultimul tip de obiecte adăugat a fost obiectul **BS_Intr**, destinat introducerii confortabile de şiruri de caractere. Problema era să "adaptăm" Unit-ul astfel încît caracteristicile moştenite să funcţioneze fără probleme. În anumite condiţii, însă, nu aceasta este cea mai potrivită cale de a-ţi atinge scopul. Din acest motiv nu trebuie să se programeze orbeşte. Dacă, de exemplu, în ierarhia de obiecte **BS_Tools** este necesar un tip de obiecte pentru introducerea de şiruri de caractere, atunci este recomandabilă analizarea în prealabil a ierarhiei existente.
Cunoscînd-o se poate găsi uşor cel mai potrivit punct de cuplare. Pe de altă parte se determină cu precizie caracteristicile noului tip de obiecte.

Dacă se observă cu atenţie **BS_Tools** se pot observa următoarele caracteristici:

În primul rînd există doi parametri **X** şi **Y** cu ajutorul cărora se stabileşte poziţia de ieşire. În al doilea rînd există diferite obiecte care declară metode virtuale. În al treilea rînd metoda specifică obiect **Init** pune la dispoziţie un constructor pentru iniţializarea metodelor virtuale. În al patrulea rînd au fost declaraţi pointeri care permit tratarea dinamică a obiectelor. Şi în al cincilea rînd pentru o tratare impecabilă au fost declaraţi şi destructorii necesari. Analiza poate continua. Pentru moment ne ajung însă aceste cinci puncte.

Ar fi de bun simţ ca parametrii pentru poziţia de ieşire să fie utilizaţi şi pentru intrări. Întrucît nu se cunoaşte însă dinainte, dacă şi în ce condiţii sînt activate metodele virtuale, acest lucru trebuie planificat dinainte. Metoda **Init** trebuie aşadar scrisă ca şi constructor. Pointerul **ReadStrPtr** trebuie să permită tratarea dinamică a obiectului obţinut. Mai trebuie verificat dacă este necesar un destructor propriu, sau dacă este suficient cel din "BS_Semn".

După aceste consideraţii preliminare putem trece la programare. La început nu ne vom atinge de Unit-ul **BS_Tools**. Vom declara un nou Unit cu numele "Inputs", pentru a semnaliza faptul că în el vor fi definite obiecte de intrare. Obiectul **BS_Intr** disponibil din **BS_Tools** nu are nici o semnificaţie pentru **Inputs**.

Obiectul mamă pentru toate obiectele de intrare este **ReadStr**. El este definit ca urmaş al obiectului **BS_String**. Pentru a nu se obţine mesaje de eroare de la compilator, nu trebuie uitată instrucţiunea Uses **BS_Tools** (linia 5).

Ne-am putea declara mulţumiţi cu acest rezultat, dar se observă imediat că nu s-a lucrat cu suficientă atenţie pentru ca şi metodele moştenite **Draw, Delete, Move, GetStr** şi **Done** să funcţioneze fără greşeală. Dacă se compară **Listingul 1** cu **Listingul 2** se observă imediat unde s-au făcut corecţii (liniile 32, 35 şi 96). Şi din nou nu ne-am atins de **BS_Tools**. Atît doar că parametrii operativi au fost transmişi obiectelor. Acum funcţionează şi metodele moştenite.

Dar acum devine vizibilă şi dilema. Atunci cînd a fost conceput **BS_Tools** nu s-a dat o consideraţie deosebită modului de tratare al octeţilor atributelor în memoria de împrospătare a imaginii. Astfel un şir (string) este mutat corect şi exact, dar nu aceleaşi lucru se întîmplă şi cu atributele sale. De data aceasta eroarea nu se află în modul de dezvoltare al ierarhiei de obiecte. Ba mai mult la planificarea lui **BS_Tools** s-au omis condiţiile de specificare ale limitelor (marginilor). În acest caz concret obiectele dezvoltate pentru ieşirea pe ecran nu funcţionează corect.

Este evidentă necesitatea de a concepe din nou Unit-ul **BS_Tools**. Erorile se regăseau mai ales în insuficienţa previziunilor privind ansamblul de caracteristici ce trebuie realizate de obiectele lui **BS_Tools**. Dacă se doreşte elaborarea unei ierarhii de obiecte, atunci trebuie dinainte ştiut ce anume trebuie să facă şi cum trebuie să facă acele obiecte. Astfel pot fi prevăzute problemele posibile şi pot fi stabilite detaliile de funcţionare.

Prima soluţie deplin valabilă este prezentată în **Listing 3**. Tipul de obiecte **BS_Intr** a fost îndepărtat. În locul acestuia este folosită o variantă îmbunătăţită a tipului **BS_Read**. Pentru o tratare corectă a atributelor în obiectul **BS_Semn** au fost introduse variabilele **ZAttr** şi **LAttr** ca parametri pentru afişarea şi ştergerea de caractere individuale. În afară de aceasta au fost adaptate corespunzător şi **Draw** şi **Delete**.

--->

```
Unit Inputs;

Interface

Uses Crt, BS_Tools;

Type
  ReadStrPtr = ^ReadStr;
  ReadStr = Object (BS_String)
    Prompt : String;
    MaxLen : Integer;
    PAttr, SAttr : Byte;
    Constructor Init (xPos,yPos : Integer;
      PrStr : String);
    Procedure SetAttr (PromptAttr, StrAttr : Byte);
    Procedure SetLen (L : Integer);
    Function Intrare (St : String): String;
  End;
Implementation
Constructor ReadStr.Init (xPos, yPos : Integer;
  PrStr : String);
Begin
  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  PAttr := Yellow + 16 * Blue;
  SAttr := PAttr;
  MaxLen := 255;
End;

Procedure ReadStr.SetAttr (PromptAttr, StrAttr : Byte);
Begin
  PAttr := PromptAttr;
  SAttr := StrAttr;
End;
Procedure ReadStr.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;

Function ReadStr.Intrare (St : String) : String;
Var
  OfsBws, SegBWS : Word;
  OldAttr : Byte;
  PozIni, PozAct, i : Integer;
  Ch : Char;
Const
  Sfirsit : Set Of Char = [ Chr(13) ];
Begin
  Intrare := St;
  GotoXY (X,Y);
  OldAttr := TextAttr;
  TextAttr := PAttr;
  Write (Prompt);
  PozIni := X + Length(Prompt);
  PozAct := PozIni;
  GotoXY (PozIni,Y);
  TextAttr := SAttr;
  Write (St);
  GotoXY (PozIni,Y);
  Repeat
    Ch := ReadKey;
    If Ch = #0 Then Begin
      Ch := ReadKey;
      Case Ch Of
        'K': {stinga}
          If PozAct > PozIni Then Dec (PozAct);
        'M': {dreapta}
          If PozAct < PozIni + MaxLen Then
            Inc (PozAct);
      End;
      GotoXY (PozAct,Y);
    End
    Else Begin
      If(PozAct - PozIni) < MaxLen Then Begin
        Write (Ch);
        Inc (PozAct);
          St := Copy(St,1,PozAct-PozIni + 1) + Ch +
            Copy(St,PozAct-PozIni+2,Length(St));
      End;
    End;
    Intrare := St;
  Until Ch In Sfirsit;
  TextAttr := OldAttr;
End;

End.
```

{Listing 1}
{Prima prezentare incorecta a Unit-ului Inputs}

```
Unit Inputs;

Interface
Uses Crt, BS_Tools;
Type
  ReadStrPtr = ^ReadStr;
  ReadStr = Object (BS_String)
    Prompt : String;
    MaxLen : Integer;
    PAttr, SAttr, OldAttr : Byte;
    Constructor Init (xPos,yPos : Integer;
      PrStr : String);
    Procedure SetAttr (PromptAttr, StrAttr : Byte);
    Procedure SetLen (L : Integer);
    Function Intrare (St : String): String;
  End;

Implementation
<---
Constructor ReadStr.Init (xPos, yPos : Integer;
  PrStr : String);
Begin
  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  Str := Prompt;
  PAttr := Yellow + 16 * Blue;
  SAttr := PAttr;
  OldAttr := TextAttr;
  MaxLen := 255;
End;

Procedure ReadStr.SetAttr (PromptAttr, StrAttr : Byte);
Begin
  PAttr := PromptAttr;
  SAttr := StrAttr;
End;
Procedure ReadStr.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;
```

```
Function ReadStr.Intrare (St : String) : String;
Var
  OfsBws, SegBWS : Word;
  PozIni, PozAct, i : Integer;
  Ch : Char;
Const
  Sfirsit : Set Of Char = [ Chr(13) ];
  Set1 : Set Of Char = [' '..'z'];
Begin
  Intrare := St;
  GotoXY (X,Y);
  TextAttr := PAttr;
  Write (Prompt);
  PozIni := X + Length(Prompt);
  PozAct := PozIni;
  GotoXY (PozIni,Y);
  TextAttr := SAttr;
  Write (St);
  GotoXY (PozIni,Y);
  Repeat
    Ch := ReadKey;
    If Ch = #0 Then Begin
      Ch := ReadKey;
      Case Ch Of
      'K': {stinga}
        If PozAct > PozIni Then Dec (PozAct);
      'M': {dreapta}
        If PozAct < PozIni + MaxLen Then
          Inc (PozAct);
      End;
      GotoXY (PozAct,Y);
    End
    Else Begin
      If(PozAct - PozIni) < MaxLen Then Begin
      If Ch In Set1 Then Begin
        Write (Ch);
        Inc (PozAct);
        St := Copy(St,1,PozAct-PozIni+1)+Ch+
          Copy(St,PozAct-PozIni+2,Length(St));
        End;
      End;
    End;
    Intrare := St;
  Until Ch In Sfirsit;
  Str := Prompt+St;
  TextAttr := OldAttr;
End;
End.
{Listing 2}
{Unit-ul Inputs adaptat pentru BS_Tools}
```

```
Unit Bs_Tools;

Interface

Uses Crt;

Const
 SegBWS = $B800; { Color}
{SegBWS = $B000;  Mono }
 OfsBWS : Word = $0000;

Terminator : Set Of Char = [Chr(0),Chr(9),
          Chr(10),Chr(12),Chr(13)];
Set1 : Set Of Char = [' '..'z'];

Type

  BS_LocPtr  = ^BS_Loc;
  BS_SemnPtr  = ^BS_Semn;
  BS_StringPtr = ^BS_String;
  BS_ReadPtr  = ^BS_Read;

  BS_Loc = Object
    X,Y : Integer;
    Procedure Init (xPos, yPos : Integer);
    Function GetX : Integer;
    Function GetY : Integer;
  End;

  BS_Semn = Object (BS_Loc)
    Semn : Char;
    ZAttr, LAttr : Byte;
    Constructor Init (xPos, yPos : Integer;
      Ch : Char; Attr : Byte);
    Procedure Draw; Virtual;
    Procedure Delete; Virtual;
    Procedure Move (xNou, yNou : Integer);
    Procedure SetLAttr (LA : Byte);
    Destructor Done; Virtual;
  End;

  BS_String = Object (BS_Semn)
    Str : String;
    Constructor Init (xPos, yPos : Integer;
      S : String; StrAttr : Byte);
    Procedure Draw; Virtual;
    Function GetStr : String;
    Procedure Delete; Virtual;
  End;

  BS_Read = Object (BS_String)
    Prompt : String;
    MaxLen : Integer;
    OldAttr : Byte;
    Constructor Init (xPos, yPos: Integer;
      PrStr : String);
    Procedure SetLen (L : Integer);
    Function Intrare (St : String): String;
  End;

Implementation

Procedure BS_Loc.Init (xPos, yPos : Integer);
Begin
  X := xPos;
  Y := yPos;
End;

Function BS_Loc.GetX : Integer;
Begin
  GetX := X;
End;

Function BS_Loc.GetY : Integer;
Begin
  GetY := Y;
End;

Constructor BS_Semn.Init (xPos, yPos : Integer;
      Ch : Char; Attr : Byte);
Begin
  BS_Loc.Init (xPos,yPos);
  Semn := Ch;
  ZAttr := Attr;
  LAttr := TextAttr;
End;

Procedure BS_Semn.Draw;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS+1] := ZAttr;
  Mem[SegBWS : OfsBWS]  := Ord(Semn);
```

```
End;

Procedure BS_Semn.Delete;
Begin
  OfsBWS := 2*(X-1)+160*(Y-1);
  Mem[SegBWS : OfsBWS] := 32;
  Mem[SegBWS : OfsBWS+1] := LAttr;
End;

Procedure BS_Semn.Move (xNou, yNou : Integer);
Begin
  Delete;
  BS_Loc.Init (xNou,yNou);
  Draw;
End;

Procedure BS_Semn.SetLAttr (LA : Byte);
Begin
  LAttr := LA;
End;

Destructor BS_Semn.Done;
Begin
End;

Constructor BS_String.Init (xPos, yPos : Integer;
      S : String; StrAttr : Byte);
Begin
  BS_Loc.Init (xPos,yPos);
  Str := S;
  ZAttr := StrAttr;
End;

Procedure BS_String.Draw;
Var i,s : Integer;
  Ch : Char;
Begin
  s := X;
  For i := 1 To Length (Str) Do Begin
    BS_Semn.Init ((i-1)+s,Y,Str[i],ZAttr);
    BS_Semn.Draw;
  End;
  X := s;
  Semn := Str[1];
End;

Function BS_String.GetSt. : String;
Begin
  GetStr := Str;
End;

Procedure BS_String.Delete;
Var i : Integer;
Begin
  i := X;
  Repeat
    OfsBWS := 2*(X-1)+160*(Y-1);
    If Chr (Mem[SegBWS : OfsBWS]) in Terminator
      Then Begin
        X := i;
        Exit;
    End;
    Bs_Loc.Init(X,Y);
    BS_Semn.Delete;
    Inc(X);
  Until X = 80;
  X := i;
End;

Constructor BS_Read.Init (xPos, yPos : Integer;
      PrStr : String);
Begin
  X := xPos;
  Y := yPos;
  Prompt := PrStr;
  Str := Prompt;
  ZAttr := Yellow + 16 * Blue;
  OldAttr := TextAttr;
  MaxLen := 255;
End;

Procedure BS_Read.SetLen (L : Integer);
Begin
  If L < 256 Then
    MaxLen := L Else MaxLen := 255;
End;

Function BS_Read.Intrare (St : String): String;
Var
  PozIni, PozAct : Integer;
  Ch : Char;
  XOld,YOld : Integer;
Begin
  XOld := X;
  YOld := Y;
  Intrare := St;
  BS_String.Init (X,Y,Prompt,ZAttr);
  BS_String.Draw;
  PozIni := X + Length(Prompt);
  PozAct := PozIni;
  BS_String.Init (PozIni,Y,St,ZAttr);
  BS_String.Draw;
  GotoXY (PozIni,Y);
  Repeat
    Ch := ReadKey;
    If Ch = ű0 Then Begin
      Ch := ReadKey;
      Case Ch Of
      'K': {stinga}
        If PozAct > PozIni Then Dec (PozAct);
      'M': {dreapta}
        If PozAct < PozIni + MaxLen Then
          Inc (PozAct);
      End;
      GotoXY (PozAct,Y);
    End
    Else Begin
      If(PozAct - PozIni) < MaxLen Then Begin
       If Ch In Set1 Then Begin
        BS_Semn.Init (PozAct,Y,Ch,ZAttr);
        BS_Semn.Draw;
        Inc (PozAct);
        GotoXY (PozAct,Y);
        St := Copy(St,1,PozAct-PozIni-1)+Ch+
          Copy(St,PozAct-PozIni+1,Length(St));
        End;
      End;
    End;
    Intrare := St;
  Until Ch In Terminator;
  Str := Prompt+St;
  TextAttr := OldAttr;
  BS_Loc.Init (XOld,YOld);
End;

End.

{Listing 3}
{Prezentarea completa a Unit-ului BS_Tools}
```

```
Program DoneTest;

Uses BS_Tools;

Var A1 : BS_ReadPtr;
  S : String;
Begin
  Str (MemAvail:8,S);
  WriteLn ('Heap liber : '+S);
  New (A1,Init(5,5,'Intrare :'));
  Dispose (A1,Done);
  Str (MemAvail:8,S);
  WriteLn ('Heap liber : '+S);
End.

{Listing 4}
{Programul permite sa se controleze corecta tratare a memoriei}
```

--->

Şi rutinele de intrare din "Inputs" (**Listing 2**) şi **BS_Tools** (**Listing 3**) trebuie comparate. Modificările de dataliu lămuresc mai buna utilizare a variabilelor anterior amintite. Au fost gîndite şi posibelele extensii. Astfel sînt prezentate tipurile de seturi de caractere Toate_Semnele, NumReal şi NumOnly.

Înainte de a continua extensia consecventă a lui **BS_Tools**, trebuie verificat dacă metoda Done a obiectului **BS_Semn** eliberează într-adevăr întreaga memorie pentru următorul obiect. Un mic program (**Listing 4**) ne va lămuri imediat. Cu toate că un obiect de tipul **BS_Semn** ocupă un alt număr de octeţi de memorie în heap, decît de exemplu un obiect de tipul **BS_Read**, metoda **Done** eliberează o cantitate corectă de memorie.

Acum se poate trece la realizarea de obiecte pentru introducerea de numere. Soluţiile corespunzătoare devin cît se poate de simple datorită puterii metodei **BS.Read_Intr**. Poate fi extins atît **BS_Tools**, dar poate fi realizat şi un alt Unit. Din motive de spaţiu vom prefera, în continuare, ultima variantă. Cu ajutorul editorului însă noul cod poate fi inserat oricînd în codul lui **BS_Tools**.

Scrierea unui nou obiect, să-i spunem **Read_Real**, nu mai pune probleme. În acesta pot fi utilizaţi doi noi parametri **Min** şi **Max** care să permită specificarea intervalului de valori în care este aşteptată specificarea valorii de intrare. Faţă de obiectul mamă, constructorul va trebui astfel modificat încît să permită introducerea de valori numerice. Prin intermediul unei proceduri, să-i spunem **SetMinMax**, aceste limite vor trebui să poată fi modificate.

Astfel se poate continua şi se pot elabora obiecte şi pentru introducerea celorlalte tipuri de valori numerice. Obiectele **ReadInt**, **ReadLongInt** sau **ReadByte** pot fi programate toate ca urmaşi ai obiectului **BS_Read**.

Dacă v-aţi chnuit pînă în acest moment şi aţi introdus în calculator Unit-ul **BS_Tools** şi probabil l-aţi mai şi extins, atunci probabil că nu veţi fi mulţumiţi cu posibilităţile de editare oferite de **BS_Read**. Nici nu ne-am putut permite în condiţiile date să încercăm să ridicăm cît se poate de mult acest confort.

Este mai bine cu OOP sau nu? Aşa, sau asemănător, îşi pun întrebarea utilizatorii care au auzit de programarea orientată obiect, dar care nu ştiu ce se ascunde în spatele ei.

Versiunile 5.5 şi 6.0 ale lui Turbo Pascal nu obligă pe nimeni să programeze orientat obiect. Şi pe mai departe se poate programa în maniera tradiţională. Este şi bine aşa.

Limbajele orientate obiect (Pascal sau C) au fost create cu scopul sporirii eficienţei şi transparenţei programării pentru a putea stăpîni mai bine complexitatea mereu sporită a programelor.

Cu ajutorul tehnicii de programare cu subprograme şi Unit-uri se puteau concepe programe frumos structurate care să rezolve sarcini complexe. La conlucrarea lor trebuia să se ţină seama de multe condiţii de limite astfel încît apăreau probleme de potriviri şi interschimbarea unor module era limitată sau chiar imposibilă.

De exemplu este foarte greu să se elaboreze Unit-uri generale şi universale pentru o suprafaţă utilizator, care să poată fi utilizată fără probleme în cele mai diferite aplicaţii. Biblioteca de obiecte Borland "Turbo-Vision" este un prim pas de a sprijini programarea unor aplicaţii complexe cu metode orientate obiect.

Utilizarea tehnicilor orientate obiect are sens mai ales la scrierea unor aplicaţii complexe. Cu ajutorul structurilor poliforme se obţine o transparenţă sporită a limbajului. În exemplul nostru **BS_Tools** dă metodele **Draw, Delete** şi **Move** pentru toţi urmaşii lui **BS_Semn**. Toate obiectele utilizează aceeaşi însărcinare pentru a realiza acţiuni diferite.

Decizia de a utiliza sau nu programarea orientată obiect depinde de aplicaţie.Dacă s-a decis utilizarea acestei tehnici moderne de programare atunci ea trebuie folosită consecvent şi cuprinzător.

(R.M.)

Programarea logică

Turbo-Prolog - Partea întîia

Limbajul de programare Prolog este un instrument excelent pentru rezolvarea de probleme logice. În prima parte, a unei serii de trei episoade, ne vom ocupa de cîteva din principiile de bază.

Numele Prolog este o prescurtare a sintagmei PROgramming în LOGic. Acest limbaj de programare a fost elaborat în 1972 de A. Comlerauer şi P. Rousel pentru scrierea de aplicaţii în domeniul inteligenţei artificale. Cu toate că atunci limbajul Lisp era greu de învăţat, la vremea respectivă a fost preferat limbajului Prolog.

În jurul anului 1980 dezvoltarea intensivă a sistemelor expert a necesitat sisteme Prolog puternice. Astfel au apărut diferite dialecte ale acestui limbaj. Pînă în ziua de astăzi nu există încă un standard unitar. Ca fundamente orientative pentru majoritatea implementărilor Prolog este folosit cvasistandardul formulat de Clocksin şi Mellish.

Programarea într-un limbaj de programare uzual, cum sînt Pascal, C, Fortran, etc., constă în principal în formularea de proceduri, funcţii şi instrucţiuni. Trebuie deci să i se spună exact calculatorului ce şi cum trebuie făcut ceva. În cazul limbajului Prolog, programatorul are doar sarcina de a descrie problema de rezolvat într-o formă apropiată limbajului natural. Calea de rezolvare este lăsată în seama sistemului Prolog, care va încerca să rezolve problema prin algoritmi mai mult sau mai puţin eficienţi. Pentru a accelera obţinerea unei soluţii, programatorul poate da unele indicaţii ajutătoare.

Limbajul Prolog poate fi utilizat pentru toate aplicaţiile care necesită deducţii formale. Problemele numerice pot fi programate mai puţin eficient, deoarece structura Prolog nu este recomandată pentru acest scop.

Sistemele expert care simulează reacţiile unui specialist prin intermediul unui mecanism deductiv, reprezintă un domeniu în care Prolog poate fi utilizat cu rezultate excelente. La acestea se adaugă sistemele informaţionale în limbaj natural, care permit şi utilizatorilor cu mai puţine cunoştinţe tehnice să descrie o problemă pentru a fi înţeleasă de calculator. Un domeniu larg de utilizare este şi cel al jocurilor şi simulărilor care necesită deducţii formale.

Turbo-Prolog conţine majoritatea caracteristicilor Prolog stabilite de Clocksin şi Mellish. Spre deosebire de sistemele Prolog anterioare care erau interpretoare, Turbo Prolog este compilator. Programele obţinute sînt astfel mai rapide decît cele obţinute pe PC-uri cu alte sisteme Prolog.

Acest compilator conţine o serie de predicate standard pentru funcţii (prelucrare şiruri, acces date, grafică, tehnica ferestrelor şi ieşiri sonore). Este posibilă cuplarea unor secvenţe de program elaborate cu un limbaj procedural. Spre deosebire de versiunile Prolog mai vechi, pentru un control mai eficient pot fi declarate variabile.

Cu toate că printre domeniile speciale ale limbajului Prolog nu se numără şi calculul numeric există şi predicate pentru lucrul cu aritmetica cu virgulă fixă şi cu virgulă flotantă. În plus spre deosebire de celelalte sisteme Prolog este disponibilă o suprafaţă utilizator tipică Turbo.

Un program Turbo Prolog poate conţine următoarele secţiuni:

- - Comenzi compilator
- - Domains
- - Database
- - Predicates
- - Goal şi
- - Clauses

Într-un program nu trebuie să fie conţinute toate aceste secţiuni dar ordinea în care au fost enumerate este obligatorie. Secţiunile individuale sînt introduse printr-un cuvînt cheie corespunzător.

Pentru a se putea verifica faptul că toate obiectele corespund tipului declarat, în partea de declaraţii "domains" trebuie declarate toate tipurile de date.

Turbo - Prolog cunoaşte următoarele tipuri standard de domenii predefinite:

char - caracter încadrat de apostroafe
integer - numere întregi între -32768 şi 32767
real - numere în virgulă fixă între 1E-307 şi 1E+308
string - şir de caractere încadrat de ghilimele
symbol - succesiune de caractere formată din litere, cifre şi liniuţe de subliniere, primul caracter trebuind să fie o literă mică.
file - nume simbolic de fişier

Faptele pot fi descrise prin relaţii simbolice. Următoarele declaraţii reprezintă fapte care pot fi înţelese de Turbo-Prolog:

```
frati (ion, petre).
barbat (alexandru).
tata (ion, alexandru).
```

În Prolog faptele mai sînt denumite şi clauze. În limba română faptele anterioare ar putea fi exprimate astfel:

Ion şi Petre sînt fraţi.

Alexandru este bărbat.
Ion este tatăl lui Alexandru.

Relaţiile clauzelor anterior enumerate sînt denumite predicate, iar parametrii clauzelor sînt denumiţi argumente. Un predicat poate conţine oricîte argumente, dar poate şi să nu conţină niciunul. În Turbo-Prolog un predicat trebuie utilizat întotdeauna cu acelaşi număr de argumente.

Fiecare clauză se sfîrşeşte cu un punct, iar argumentele ei sînt separate cu virgule. Numele de predicate încep întotdeauna cu literă mică. Celelalte caractere pot fi atît litere mici cît şi litere mari, cifre sau linii de subliniere.

În secţiunea predicates trebuie definite toate predicatele. Pentru a declara un predicat trebuie specificate numele său şi tipul argumentelor sale:

```
predicates
  data_nasterii (symbol,
    integer, integer,integer)
```

Predicatul anterior ar fi fost mai inteligibl dacă pentru fiecare argument s-ar fi definit un tip de domeniu explicit:

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
```

Într-un program Prolog pot apare mai multe secţiuni predicates. În secţiunea clauses sînt specificate clauzele şi regulile pentru predicatele anterior definite.

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
clauses
  data_nasterii (ion, 15, 5, 1969).
  data_nasterii (gelu, 26, 10, 1940).
  data_nasterii (vasile, 11, 9, 1969).
```

Clauzele şi regulile constituie aşa-numita bază de date (database) a unui program Prolog, căreia i se pot adresa întrebări în timpul execuţiei.

Dacă programul de mai sus este lansat în execuţie cu comanda RUN, sistemul cere introducerea unui goal extern, aceasta însemnînd că programatorul trebuie să pună o întrebare sistemului Prolog.

De exemplu:

```
Goal: data_nasterii (ion, 15, 5, 1969)
```

Turbo-Prolog răspunde cu True.

De modul în care ajunge sistemul la o soluţie ne vom ocupa mai tîrziu. Un goal poate fi inserat şi direct în interiorul unui program, caz în care vorbim despre un goal intern.

Într-o clauză sau într-un goal pot fi utilizate şi variabile pentru a specifica relaţiile unui obiect. Variabilele necesită rezervare de spaţiu iar numele lor trebuie să înceapă cu o literă mare putînd continua cu litere, cifre sau linii de subliniere.

Un nume de variabilă nu poate conţine mai mult de 250 de caractere şi pentru o mai bună lizibilitate trebuie să facă o referire la obiectul reprezentat. Prin utilizarea variabilelor într-un goal devine posibilă interogarea de informaţii din baza de date şi nu doar obţinerea unor răspunsuri da-nu.

Cu ajutorul goal-ului următor, din baza de date anterior creată pot fi interogate anumite date de naştere.

```
Goal: data-nasterii (Nume,Zi,Luna,An)
```

Sistemul va răspunde astfel:

```
Nume=ion,Zi=15,Luna=5,An=1969
Nume=gelu,Zi=26,Luna=10,An=1940
Nume=vasile,Zi=11,Luna=9,An=1969
3 Solutions
```

De îndată ce sistemul Prolog atribuie o valoare unei variabile (de ex. Nume = ion), vorbim despre ea ca fiind o variabilă legată. Dacă nu i-a fost atribuită încă o valoare, atunci vorbim despre o variabilă liberă. La lansarea în execuţie a programului toate variabilele sînt libere.

Uneori la o interogare destinată unui anumit scop unele argumente sînt lipsite de semnificaţie. Acestea pot fi înlocuite, în acest caz, prin variabile anonime. Acestea sînt reprezentate prin intermediul unei linii de subliniere şi nu li se atribuie o valoare în procesul de soluţionare.

Dacă programatorul doreşte să afle în ce an sînt născute persoanele din baza de date, atunci va trebui să formuleze următorul goal:

```
Goal: data_nasterii (Nume,_,_,An)
```

Sistemul va răspunde, în acest caz, cu:

```
Nume = ion, An = 1969
Nume = gelu, An = 1940
Nume = vasile, An = 1969
3 Solutions
```

Pînă acum relaţiile dintre obiecte au fost date prin clauze. În continuare vom extinde baza de cunoştinţe. Prin intermediul regulilor, Turbo Prolog este în stare să deducă noi clauze din cele deja existente în baza de cunoştinţe. O regulă este o expresie prin care se stabileşte valoarea de adevăr a unei clauze speciale în funcţie de una sau mai multe alte clauze. Ea constă dintr-un aşa-numit antet de regulă, urmat de cuvîntul **"if"**, sau de **":-"**. La urmă sînt exprimate toate condiţiile de care depinde luarea deciziei. Condiţiile sînt cuplate între ele prin operatori **"şi"** respectiv **"sau"**.

O cuplare **şi** este reprezentată prin "and" sau ",", iar una **sau** prin "or" sau ";". Punctul semnifică sfîrşitul. Cu următoarea regulă se poate stabili, de exemplu, dacă două persoane s-au născut în acelaşi an:

```
acelasi_an_nastere (Nume1,Nume2):-
data_nasterii (Nume1,_,_,An),
data_nasterii (Nume2,_,_,An).
```

Întregul program va arăta astfel:

```
domains
  nume = symbol
  zi, luna, an = integer
predicates
  data_nasterii (nume, zi, luna, an)
  acelasi_an_nastere (nume, nume)
clauses
  data_nasterii (ion, 15, 5, 1969).
  data_nasterii (gelu, 26, 10, 1940).
  data_nasterii (vasile, 11, 9, 1969).
  acelasi_an_nastere (Nume1,Nume2):-
    data_nasterii (Nume1,_,_,An),
    data_nasterii (Nume2,_,_,An).
```

Acum se poate pune, de exemplu, întrebarea:

```
Goal: acelasi_an_nastere (ion, X)
```

Sistemul va da următorul răspuns:

```
X = ion
X = vasile
2 Solutions
```

Mecanismul de transmitere a valorilor se realizează în Prolog prin unificare. În acest punct acest limbaj de programare se diferenţiază de aşa-numitele limbaje procedurale. Pentru a înţelge limbajul Prolog, este de o importanţă hotărîtoare înţelegerea mecanismului de unificare. El descrie o procedură prin care limbajul Prolog încearcă realizarea unei corespondenţe între un termen şi o clauză sau un antet de regulă, pentru atingerea unui scop.

Un termen poate fi unificat cu alt termen (poate fi echivalat) dacă:

1) ambii termeni apar în predicate cu acelaşi număr de argumente, în aceeaşi poziţie;
2) ambii termeni sînt de tipuri echivalente (char poate fi echivalent numai cu char, ş.a.m.d.)
3) subtermenii corespunzători pot fi de asemenea echivalaţi.

În cadrul unificării sînt valabile următoarele:

1) variabilele libere pot fi unificate cu orice termen care satisface condiţiile mai sus menţionate. Prin unificare variabila este legată de valoarea termenului.
2) constantele pot fi unificate cu ele însele, sau cu variabile libere. Dacă o variabilă liberă este echivalată cu o constantă, atunci va lua valoarea acelei constante.
3) variabilele libere pot fi unificate cu oricare alte variabile. După echivalare ambele vor fi tratate ca o singură variabilă. Dacă în cele din urmă una din cele două este legată de o valoare, atunci şi cealaltă va fi instanţiată cu această valoare.
4) predicatele pot fi echivalate dacă posedă acelaşi nume de relaţii, numărul argumentelor este acelaşi şi perechile de argumente pot fi unificate între ele.

În continuare unificarea va fi explicitată prin intermediul unui exemplu:

```
domains
  loc, persoana = symbol
  an = integer
  locul_nasterii = născut (loc,an)
predicates
  date_de_nastere (persoana,locul_nasterii)
clauses
  date_de_nastere (ion,
        nascut(calarasi,1964)).
date_de_nastere (ana,
        nascut(zimnicea,1968)).
```

Dacă se specifică acum goal-ul extern:

```
Goal: date_de_nastere (X,Y)
```

atunci sistemul Prolog încearcă să unifice clauzele din goal cu prima clauză din program. Acest lucru este posibil deoarece atît X cît şi Y sînt variabile libere şi pot fi deci legate de orice alt termen al aceluiaşi predicat.

În cazul celei de-a doua clauze, din aceleaşi motive, echivalarea este de asemenea posibilă. Sistemul va furniza deci următorul răspuns:

```
X = ion, Y = nascut(calarasi,1964)
X = ana, Y = nascut(zimnicea,1968)
2 Solutions
```

În cele din urmă să punem sistemului următoarea întrebare:

```
date_de_nastere (X,nascut(zimnicea,Y))
```

Din nou sistemul încearcă o unificare cu prima clauză. Astfel variabila X este legată mai întîi de valoarea "ion" şi apoi se încearcă unificarea lui "(calarasi, 1964)" cu "(zimnicea, 1968)".

Cum am remarcat deja, unificarea termenilor este posibilă numai atunci cînd ambii au acelaşi nume de relaţie, acelaşi număr de argumente şi în plus toţi subtermenii pot fi echivalaţi.

În exemplul anterior deşi ambii termeni au acelaşi nume de relaţie, "născut", şi acelaşi număr de argumente, constanta "calarasi" poate fi unificată numai cu ea însăşi sau cu o variabilă liberă; deci în acest caz unificarea nu este posibilă. Procedura de unificare eşuează deci la prima clauză, urmează cea de-a doua. În cazul ei vor fi comparaţi termenii "date_de_nastere (X,nascut(zimnicea,Y))" şi "date_de_nastere (ana, nascut(zimnicea,1968))". Variabila X va fi instanţiată cu valoarea ana. Echivalarea celor doi termeni "nascut(zimnicea,Y)" şi "nascut(zimnicea,1968)" are loc cu succes, deoarece este utilizat acelaşi nume de relaţie "nascut" şi numărul argumentelor este acelaşi.

Constanta "zimnicea" este unificată cu ea însăşi, iar variabila Y este instanţiată cu valoarea 1968. Astfel se va obţine soluţia:

```
X = ana, Y = 1968
1 Solution
```

Noţiunea backtracking desemnează o strategie de căutare specifică. Dacă pe calea pe care s-a pornit nu se ajunge la o soluţie, atunci programul se întoarce şi caută în altă direcţie.

Prin backtracking sînt găsite toate soluţiile unei probleme. Cea mai bună comparaţie se poate face cu căutarea într-un labirint. Pentru a se face o deplasare dintr-un loc într-alt loc se urmăreşte un drum exact, atît timp cît nu există bifurcaţii. Dacă se ajunge la un loc în care există mai multe alternative,

atunci se selectează una dintre posibilităţi. Dacă se ajunge într-o fundătură, atunci se întoarce la punctul de bifurcare şi se alege un drum care n-a fost încercat încă (ş.a.m.d.). Cu această metodă se poate atinge în orice caz scopul, în cazul în care un astfel de drum există (se poate întîmpla şi să nu existe un astfel de drum).

Noţiunea backtracking desemnează revenirea la acea poziţie în care existau mai multe variante, pentru a porni de acolo pe un alt drum.

Cum este utilizată această metodă în Turbo Prolog va fi înţeles cu ajutorul exemplului următor:

```
domains
   persoana, sport = symbol
predicates
   practica (persoana, sport)
clauses
   practica (gicu, tenis).
   practica (ion, volei).
   practica (radu, tenis).
   practica (nicolae, X) if
   practica (ion, X).
```

Pentru a determina toate persoanele care practică acelaşi sport va trebui specificat următorul goal:

```
Goal: practica (Persoana1,Sport),
      practica (Persoana2,Sport),
      Persoana 1 <> Persoana 2.
```

Turbo - Prolog ajunge la soluţie cu ajutorul backtracking-ului şi unificării pe următoarea cale:

- - sistemul porneşte de la primul subgoal: "practica (Persoana1,Sport)"
- - este cercetată prima clauză, este posibilă unificarea "Persoana1 = gicu" şi "Sport = tenis".
- - se trece apoi la al doilea subgoal "practica (Persoana2,Sport)". Variabila "Sport" este deja legată de valoarea "tenis" ceea ce duce la transformarea celui de-al doilea subgoal în "practica (Persoana2, tenis)". Variabila "Persoana2" mai este încă liberă.
- - la depistarea celui de-al doilea scop Prolog porneşte din nou de la prima clauză. Se reuşeşte unificarea "Persoana2 = gicu".
- - acum se încearcă verificarea şi a celei de-a treia condiţii "Persoana1 <> Persoana2". Deoarece ambele variabile sînt legate de valoarea "gicu", condiţia nu se verifică şi scopul nu poate fi atins cu această legare a variabilelor.
- - cu aceasta ultima legare de variabilă ("Persoana2 = gicu") este din nou dezlegată şi sistemul încearcă unificarea celui de-al doilea subgoal cu cea de-a doua clauză. Acest lucru nu reuşeşte deoarece variabila "Sport" este legată încă de valoarea "tenis".
- - din această cauză se trece la a treia clauză la care se reuşeşte unificarea "Persoana2 = radu". Acum se verifică din nou al treilea subgoal. Deoarece în acest caz "Persoana1 = gicu" <> "Persoana2 = radu" şi deci condiţia este satisfăcută, s-a găsit o soluţie la interogare ("gicu, tenis, radu"). Aceasta va fi şi afişată.
- - Deoarece în cazul goal-urilor externe, Turbo-Prolog caută toate soluţiile posibile, prelucrarea programului nu ia încă sfîrşit. Ultima atribuire de variabilă ("Persoana2 = radu") este desfăcută.
- - sînt examinate regulile pentru a patra clauză. Sistemul încearcă s-o acopere cu al doilea subgoal. Pentru a verifica "practica (gicu, tenis)" trebuie să corespundă cu "practica (ion, tenis)". Acest lucru nu este posibil şi deci nu există o a doua soluţie a acestui subgoal.
- - deoarece au fost încercate toate posibilităţile pentru al doilea subgoal, sînt dezlegate (resetate) şi variabilele primului subgoal şi urmează unificarea primului scop cu cea de-a doua clauză.
- - acest lucru reuşeşte cu "Persoana1 = ion" şi "Sport = volei". Al doilea subgoal se transformă deci în "practica (Persoana2, volei)". Unificarea cu prima clauză nu este posibilă, echivalarea cu cea de-a doua clauză este posibilă, dar nu este satisfăcută cea de a treia condiţie ("Persoana1 <> Persoana2"). Unificarea cu a treia clauză nu este nici ea posibilă.
- - mai rămîne de examinat a patra clauză. Pentru a verifica "practica (nicolae,volei)" trebuie să se verifice "practica (ion,volei)". Acest lucru este valabil prin cea de a doua clauză şi cum şi cea de-a treia condiţie este satisfăcută ("Persoana1 = ion" <> "Persoana2 = nicolae") se obţine o nouă soluţie ("ion, volei, nicolae"), care va fi de asemenea afişată.
- - în mod analog primul goal va fi unificat cu cea de a treia şi apoi cu cea de a patra clauză. Apoi sistemul va căuta soluţii prin observarea celorlalte subgoal-uri.

Soluţia finală va fi:

```
Persoana1 = gicu, Sport = tenis,
    Persoana2 = radu
Persoana1 = ion, Sport = volei,
    Persoana2 = nicolae
Persoana1 = radu, Sport = tenis,
    Persoana2 = gicu
Persoana1 = nicolae, Sport = volei,
    Persoana2 = ion
```

Comanda "trace" permite execuţia pas cu pas. Mesajele ferestrei trace au următoarea semnificaţie:

Call: apelul predicatelor. Sistemul încearcă să îndeplinească scopul clauzei.

Return: clauza a putut fi îndeplinită, sistemul trece la următorul subgoal.

Redo: backtracking pentru a permite şi alte soluţii sau dacă nu s-a obţinut o soluţie cu valorile momentane ale variabilelor.

Fail: scopul nu poate fi realizat.

În partea următoare a cursului de programare logică ne vom ocupa de alte principii de bază. Pe lîngă fundamentele limbajului de programare Prolog va fi dat ca exemplu şi modul de rezolvare al unei probleme logice.

(R.M.)

Arta de a împărţi

Partiţionarea harddisk-ului

Aproape fiecare posesor de PC cu harddisk a făcut cel puţin odată o partiţionare, dar foarte puţini ştiu exact, ce este şi la ce foloseşte.

Cine nu doreşte să-şi stocheze datele pe harddisk efectuînd de unul singur gestiunea sectoarelor (care date pe care sector se află - la un disc de 20 MByte, aproximativ 40.000 de sectoare de 512 Byte....), poate lăsa aceasta sub controlul şi administrarea unui sistem de operare. Dar înainte ca acesta să-şi intre în rol, harddiskul trebuie pregătit corespunzător: el trebuie partiţionat şi formatat corect.

Trebuie definite toate zonele de pe disc care sînt la dispoziţia sistemului. Dacă, în plus, la pornirea calculatorului am vrea să pornească direct de pe harddisk, este neapărată nevoie de "încărcătorul de partiţii". Acesta este un program micuţ fără de care pornirea "automată" a calculatorului nici nu ar fi posibilă.

Partiţia este un domeniu continuu de pe harddisk, care fizic reprezintă toate sectoarele dintre două adrese de harddisk. O adresă de harddisk este formată dintr-un triplet de valori şi anume numărul de cilindru, numărul de cap şi numărul de sector. Sectoarele de pe o pistă se numerotează totdeauna începînd cu "1", iar capetele de citire-scriere precum şi cilindri se numerotează totdeauna începînd cu 0 şi terminînd cu (numărul_maxim -1).

Hotărîtor în partiţionarea harddisk-ului este aşa-numitul "articol de partiţionare". El se găseşte pe aproape toate harddisk-urile şi ocupă întotdeauna primul sector (absolut) de date (sectorul 1, cilindrul 0, capul 0). Trebuie spus că acesta este singurul articol de pe fiecare harddisk, a cărui poziţie este cunoscută. Poziţia tuturor celorlalte date, (de exemplu cea a sectorului de boot, a tabelei de alocare şi a fişierelor) este variabilă. Poziţia lor depinde direct, mai mult sau mai puţin, de datele din "articolul de partiţionare".

Mărimea standard a sectorului la harddisk-urile PC-urilor compatibile IBM este de 512 Byte. În ciuda acestei dimensiuni, mai degrabă mici, toate datele necesare articolului de partiţionare pot fi depuse uşor pe un singur sector.

După cum se poate observa din figura 1, articolul de partiţionare se compune din 3 părţi. La început se află încărcătorul de partiţii, la deplasamentul constant 446 (1BC hex) se află tabela de partiţii (de 64 Byte) iar la sfîrşit se află marcajul de date de sistem (2 Byte - anume "55AA" hex).

Acest marcaj nu se găseşte doar în articolul de partiţionare ci şi în articolul de boot, şi ocupă întotdeauna ultimii doi octeţi.

Graţie acestui marcaj, programele sînt în stare să determine repede şi simplu, dacă datele din cadrul acestui sector sînt - sau nu - interpretabile conform aşteptărilor.

Dacă de pildă la un articol de partiţionare, altfel bun, se şterge acest marcaj , atunci, după pornirea sistemului de pe dischetă, sistemul de operare DOS nu poate accede harddisk-ul . Nici pornirea sistemului de pe harddisk nu se poate face în aceste condiţii.

Încărcătorul de partiţii este un program în limbaj maşină, lung de aproximativ 220 octeţi, care joacă

Adresa harddisk: Cilindru 0, Pista 0, Sector 1						Format hexa
Offset 0						Încărcătorul de partiţii
FA33C08E	D0BC007C	8BF45007	501FFBFC	BF0006B9	0001F2A5	Codul încărcătorului
EA1D0600	00BEBE07	B304803C	80740E80	3C00751C	83C610FE	de partiţii çu mesaje
CB75EFCD	188B148B	4C028BEE	83C610FE	CB741A80	3C0074F4	de eroare în format
BE8B06AC	3C00740B	56BB0700	B40ECD10	5EEBF0EB	FEBF0500	ASCII.
BB007CB8	010257CD	135F730C	33C0CD13	4F75EDBE	A706EBD3	(lung. cca. 230 byte)
BECD06BF	FE7D813D	55AA75C7	8BF5EA00	7C00 0 U	ngul	
tige	Par	titi	onst	abel	le0F	
ehle	r be	im L	aden	des	Bet	
rieb	ssys	tems	0Kei	n Be	trie	
bssy	stem	vor	hand	en00	0000	Domeniul neutilizat
0000	0000	0000	0000	0000	0000	este completat cu 0
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	0000	0000	0000	
0000	0000	0000	00			
Offset 446						Tabela de partiţii
80010100	063F2027	20000000	E03F0100			4 intrări în tabelă
00000128	053F2072	00400100	00580200			a cîte 16 byte.
0000	0000	0000	0000			
0000	0000	0000	0000			
Offset 510						Identificator sistem
55AA						

Fig. 1. Alcătuirea unui articol de partiţionare de 512 Byte, în format hexa

un rol scurt, dar important, la fiecare pornire a calculatorului de pe harddisk (pornire "la rece" sau "la cald"). O descriere generală a pornirii sistemului va ajuta la o înţelegere mai bună a fenomenului.

De fiecare dată cînd un PC este pornit sau cînd s-a apăsat tasta RESET, calculatorul se iniţializează. El se pune deci într-o stare iniţială definită. Pentru aceasta procesorul execută rutinele POST (Power On Self Test) memorate în ROM. POST este o componentă fixă, care se găseşte, pe toate PC-urile, în BIOS (Basic Input Output System) şi răspunde de faptul ca toate unităţile externe să fie iniţializate şi aduse în stare operaţională (din punct de vedere al procesorului). Rutinele acestea cunosc toate cartelele de comandă (de exemplu "Timer", "Controller de harddisk", "Controller de întreruperi" etc.). Ele verifică funcţionalitatea fiecărei componente pentru ca apoi să le aducă într-o stare iniţială definită.

În momentul în care calculatorul este pe deplin operaţional, ceea ce înseamnă că toate resursele (cum ar fi RAM, harddisk-ul sau interfeţele) sînt operaţionale, intră în acţiune aşa-numitul "încărcător BOOTSTRAP". Dacă în acest moment în unitatea A se află o dischetă, acesta citeşte primul sector de pe ea (0,0,1) în memoria de bază (la adresa "7C00 hex") şi predă controlul asupra procesorului, fără nici o altă verificare, codului citit de pe disc. Dacă este vorba de articolul de boot al unei dischete sistem, atunci acest sector porneşte procedura de încărcare, iar încărcătorul de partiţii nu mai intră în acţiune. Deoarece dischetele nu pot fi partiţionate, primul lor sector va conţine întotdeauna articolul de boot. Deci, din punct de vedere logic, o dischetă este de fapt o partiţie de sine stătătoare, conţinînd toate componentele specifice necesare sistemului de operare, dintre care primul este totdeauna articolul de boot.

Dacă în unitate nu se află nici o dischetă, încărcătorul de BOOTSTRAP citeşte articolul de partiţionare în memoria de bază (la adresa sa fixă "7C00 hex") şi predă încărcătorului de partiţii controlul asupra procesorului. Acesta, mai întîi, se copiază pe el însuşi, cu tot cu tabela de partiţii şi marcajul de date sistem de la adresa "7C00 hex" la adresa "0600 hex" (în memoria de bază), astfel încît zona tampon de la adresa "7C00 hex"să se elibereze pentru alte date. În continuare, el caută în tabela de partiţii (care se află la sfîrşitul articolului de partiţionare), o aşa-numită "partiţie activă". În momentul în care în tabelă găseşte o înregistrare cu caracteristicile căutate, încărcătorul de partiţii extrage de aici adresa de început (pe harddisk) a partiţiei respective, preluînd exact un sector de la acea adresă. Acest sector va fi depus în memoria de bază, la adresa "7c00 hexa". Dacă acest sector conţine marcajul de date sistem, atunci încărcătorul de partiţii presupune că este vorba despre articolul de boot al partiţiei şi îi predă acestuia controlul asupra procesorului.

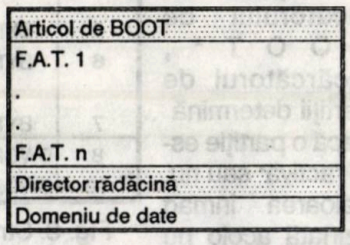


Fig. 2. Construcţia schematică a unei partiţii DOS

Dacă marcajul de date sistem lipseşte sau nu există partiţie activă în tabelă, atunci încărcătorul de partiţii dă un mesaj corespunzător la consolă (vezi şi fig. 1) şi apoi "opreşte" calculatorul (lăsînd procesorul să execute o buclă infinită).

Predînd controlul articolului de boot al partiţiei active din tabela de partiţii, încărcătorul de partiţii iniţiază încărcarea sistemului de operare care se află în cadrul acelei partiţii. În cadrul partiţiei se foloseşte totdeauna un tip specific de gestiune a datelor, caracteristic unui anumit sistem de operare.

Din momentul în care articolul de boot obţine controlul, între pornirea sistemului de operare de pe dischetă şi pornirea de pe harddisk nu există diferenţe principiale. Funcţie de datele înglobate în el, stabileşte poziţia directorului rădăcină şi verifică dacă există acolo cele două fişiere constitutive ale sistemului de operare, IBMBIO.COM şi IBMDOS.COM (respectiv, la MS-DOS, fişierele corespunzătoare: IO.SYS şi MSDOS.SYS). Dacă le găseşte pe amîndouă, atunci articolul de boot calculează poziţia de început a zonei de date a partiţiei pe harddisk şi încarcă de acolo fişierul IBMBIO.COM în memoria de bază. Acesta conţine administrarea fişierelor, încarcă IBMDOS.COM, prelucrează comenzile din CONFIG.SYS şi încarcă la sfîrşit procesorul de comenzi COMMAND.COM. Acesta la rîndul său execută comenzile din fişierul AUTOEXEC.BAT, după care, în fine, pe ecran se afişează prompter-ul obişnuit - sistemul a pornit.

Şi fişierul IBMBIO.COM citeşte articolul de partiţionare în memoria de bază, parcurgînd tabela de partiţii cu scopul de a disponibiliza, prin intermediul sistemului de operare, toate unităţile logice DOS (începînd cu C:) .

Nu trebuie să mire faptul că primele două înregistrări din directorul rădăcină al unei partiţii DOS încărcabile conţin întotdeauna date referitioare la cele două fişiere amintite ale sistemului de operare şi că ele se află întotdeauna la începutul domeniului de date. Motivul este că articolul de boot, avînd de regulă doar 512 octeţi, poate calcula, cu ajutorul datelor sale despre partiţie, poziţia exactă de început a directorului rădăcină şi a zonei de date, dar, datorită mărimii sale reduse, el nu poate să găsească direct înregistrări de fişiere în directoare sau chiar fişiere din cadrul partiţei.

Pentru completare vom detalia pe scurt structura unei partiţii DOS. Exceptînd raportul de mărimi al celor patru componente importante, fiecare partiţie DOS are aceeaşi structură (vezi fig. 2):

- articolul de boot
- tabela de alocare a fişierelor
- directorul rădăcină
- zona de date

Primul este întotdeauna articolul de boot, care ocupă, de regulă, un sector. El este urmat de tabela de alocare a fişierelor (File Allocation Table - FAT) şi copiile de siguranţă ale acesteia. FAT reprezintă un fel de plan de ocupare pentru administrarea fişierelor, iar mărimea ei depinde direct de mărimea totală a partiţiei. Directorul rădăcină ocupă de regulă 32 de sectoare la harddisk-uri, oferind astfel loc pentru 512 înregistrări de fişiere (sau subdirectoare). În cazul dischetelor, de regulă, numărul înregistrărilor din fişierul rădăcină este mai mic.

În sfîrşit, zona de date cuprinde atît toate fişierele cît şi toate subdirectoarele (care, sub DOS, sînt administrate ca nişte fişiere).

Să revenim acum la cea mai importantă parte a articolului de partiţionare: tabela de partiţii. În general ea are o mărime de 64 Byte şi se află întotdeauna imediat înaintea marcajului de date sistem, la sfîrşitul înregistrării. Tabelul oferă loc pentru exact patru înregistrări a 16 octeţi. Fiecare înregistrare reprezintă datele cadru despre o partiţie închisă în sine. Acest lucru înseamnă că un singur harddisk poate găzdui pînă la patru sisteme de operare diferite, fiecare cu propria lui administrare de date. Prin simpla mutare a indicatorului de "partiţie activă" de la o înregistrare la alta în tabelă, încărcătorul de partiţii va fi determinat să încarce, la o nouă pornire a sistemului, sistemul de operare din partiţia nou aleasă. Se presupune că partiţia este boot-abilă, deci că se poate încărca un sistem de operare de acolo.

Fig. 3 arată structura unei înregistrări din tabelă, identică pentru toate înregistrările de acest fel. Fiecare înregistrare este compusă din următoarele componente (în prima coloană - spaţiul ocupat):

- Indicatorul de BOOT (1 Byte)
- Adresa de început a partiţiei (3 Byte)
- Indicatorul de sistem (1 Byte)
- Adresa de sfîrşit a partiţiei (3 Byte)
- Sectoare rezervate (4 Byte) - Lungimea în sectoare a partiţiei (4 Byte)

Offset	Lung.	Semnificaţie
0	BYTE	Flag-ul partiţiei active
		Bit 7 = 0 : Înregistrare de partiţie inactivă
		Bit 7 = 1 : Înregistrare de partiţie activă
		Adresa de început a partiţiei (primul sector)
1	BYTE	Bit 0 - 7 : Cap (0 - 255)
2	BYTE	Bit 0 - 5 : Sector (1 - 63)
		Bit 6 & 7 : Biţii superiori ai nr. de cilindru
3	BYTE	Bit 0 - 7 : Cilindru (0 - 1023)
4	BYTE	ID-ul sistemului de operare
		Adresa de sfîrşit a partiţiei (ultimul sector)
5	BYTE	Bit 0 - 7 : Cap (0 - 255)
6	BYTE	Bit 0 - 5 : Sector (1 - 63)
		Bit 6 & 7 : Biţii superiori ai nr. de cilindru
7	BYTE	Bit 0 - 7 : Cilindru (0 - 1023)
8	DWORD	Nr. de sectoare dinaintea partiţiei
12	DWORD	Nr. de sectoare din partiţie

Fig. 3. Structura unei intrări din tabela de partiţii

Cu ajutorul "indicatorului de BOOT", încărcătorul de partiţii determină dacă o partiţie este "activă" sau nu. Valoarea înmagazinată acolo nu conţine însă doar o informaţie binară, ci determină în plus şi unitatea de disc pe care se află partiţia.

Pentru o mai bună înţelegere, să explicăm următoarea stare de lucruri: toate accedările asupra unităţilor de harddisk şi de dischetă se fac întotdeauna prin BIOS, printr-o interfaţă definită. Aceasta este integrată în calculator deja de producătorul calculatorului, sub forma unor componente ROM.

Pentru a adresa un anumit sector sau o grupă de sectoare (pe harddisk sau pe dischetă) sînt necesari întotdeauna 5 octeţi. Un octet este folosit pentru determinarea unităţii, unul determină mărimea blocului (în sectoare) şi restul de 3 octeţi sînt ocupaţi cu informaţiile despre cilindru, cap şi sector.

Unităţile de dischetă se adresează întotdeanua începînd cu "0". Unitatea A este adresată deci cu valoarea "0" şi unitatea B cu "1". Unităţile de hardisk - din care pot exista mai multe pe un calculator - se adresează întotdeauna începînd cu "128" ("80 hex"). Un al doilea harddisk ar putea fi adresat deci prin codul de unitate "129" ("81 hex").

Încărcătorul de partiţii distinge, în forma sa uzuală, doar valorile "0" (nu este partiţia activă) şi "80 hex" (partiţie activă de pe harddisk-ul "1"), dar foloseşte indicatorul de BOOT pentru încărcarea articolului de BOOT al respectivei partiţii. Astfel, cu cîteva modificări în codul încărcătorului de partiţii, fără modificări hard de configuraţie, este posibilă încărcarea unui sistem de operare de pe oricare din harddisk-urile instalate.

Adresa de început a partiţiei descrie, sub forma unei combinaţii număr cilindru + număr cap + număr sector, adresa disc la care începe partiţia. Informaţiile conţinute în cei 3 octeţi nu sînt separate în întregime şi corespund formatului în care trebuie să fie predate interfeţei BIOS pentru încărcarea articolului de BOOT. Primul octet conţine numărul de cap, următoarele două numărul cilindrului şi al sectorului. Cei doi biţi superiori ai numărului de cilindru - format din 10 cifre binare - (este vorba aici de sistemul "dual") se regăsesc în biţii 6 şi 7 ai octetului numărului de sector, (numărul de sector fiind format din 6 cifre binare). Se vede de aici că se pot adresa prin intermediul BIOS-ului incorporat harddisk-uri cu o di-

mensiune maximă de 256 de capete (2^8), 1024 cilindri (2^{10}) şi 64 de sectoare pe pistă (2^6).

Indicatorul de sistem descrie pe un octet tipul partiţei, determină deci cărui sistem de operare respectiv cărui tip de gestiune a datelor îi aparţine respectiva partiţie. Valorile uzuale pentru partiţiile DOS sînt: 1 = partiţie DOS < 16 Mbyte, cu administrare FAT de 12 biţi
4 = partiţie DOS între 16 şi 32 Mbyte şi cu administrarea FAT de 16 biţi
5 = partiţie DOS extinsă (vezi mai jos) şi
6 = partiţie DOS > 32 Mbyte cu administrare FAT de 16 biţi

Fiecare sistem de operare (CP/M, OS/2, Xenix, etc.) are identificatorul său specific, cu ajutorul căruia poate determina foarte repede, în faza de BOOT, dacă partiţia respectivă va fi sau nu recunoscută (şi odată cu ea administrarea de fişiere aferentă) şi dacă va fi preluată în rîndul unităţilor logice disponibile.

Dacă aruncăm o privire pe articolul de partiţionare cu ajutorul lui "Partiton Table Editor" din "Norton Utilities" şi apăsăm de mai multe ori pe tasta <SPAŢIU> în cîmpul BOOT, se pot vizualiza toate tipurile de sistem posibile, în clar, cu numerele aferente în modul HEX.

Adresa de sfîrşit a partiţiei indică ultimul sector ce face parte din partiţia respectivă şi corespunde întrutotul, ca structură, adresei de început a partiţiei. Cîmpul de 4 octeţi folosit pentru numărul de sectoare rezervate conţine o valoare de 32 de biţi (dublu cuvînt) care indică cîte sectoare se află între începutul harddisk-ului (adresa 0,0,1) şi adresa de start a partiţiei. Şi lungimea partiţiei este dată de un dublu cuvînt care indică cîte sectoare conţine partiţia respectivă. În fond această informaţie şi adresa de sfîrşit a partiţiei ar fi redundante, dar ea foloseşte programelor care au nevoie de această informaţie, scutindu-le de conversii de la un format de date la altul.

În mod normal, după pornirea sistemului de operare de pe harddisk, ne aflăm pe unitatea logică C şi acolo în directorul rădăcină (vizualizat pe ecran prin prompterul "C: "). Dacă sistemul este boot-at de pe unitatea de dischetă, atunci unitatea A: devine unitatea logică curentă (aşa-numitul "current drive" în terminologia DOS).

Mai mult: cele mai multe sisteme "cunosc" unităţi logice suplimentare. Unitatea B este de obicei cea de-a doua unitate de dischete, iar harddisk-ul poate fi partiţionat, la rîndul lui, în mai multe unităţi (logice!). Acestea se selectează la pornirea sistemului de operare şi se denumesc în ordine crescătoare, începînd cu simbolul "D:", "E:", "F:" şi aşa mai departe.

Modul în care aceste unităţi logice sînt reprezentate pe harddisk şi legătura lor cu partiţionarea va fi obiectul ultimei părţi a articolului. Partiţia de boot devine întotdeauna unitatea logică C şi ea nu poate cuprinde şi alte unităţi logice. Toate celelalte unităţi logice nu se află într-o partiţie proprie, ci sînt adunate, din prisma articolului de partiţionare, în aşa-numita partiţie extinsă.

Deci, un sistem "curat" DOS nu va avea nevoie niciodată de mai mult de două înregistrări în tabela de partiţii. Indiferent de numărul de unităţi logice care există în afară de C, ele se vor găsi toate într-o singură partiţie, deci într-un domeniu de disc continuu. Din punct de vedere fizic, unităţile logice de disc sînt zone de disc succesive, care "intră" exact în limitele partiţiei extinse.

Unităţile logice începînd cu D: se pot imagina ca un şir de partiţii (extinse) încapsulate, fiecare partiţie extinsă fiind formată (cu excepţia ultimei unităţi logice) din domeniul de harddisk aferent unităţii logice şi o altă partiţie extinsă. Fiecare partiţie extinsă posedă un "articol de partiţie extinsă" propriu, structura sa fiind identică cu cea a articolului de partiţionare.

Fiecare dintre aceste articole de partiţionare conţine o înregistrare , ce caracterizează domeniul de disc aferent unităţii logice şi tipul de gestiune a fişierelor aferent ei.

Atîta timp cît nu este vorba despre ultima unitate logică, există o înregistrare suplimentară pentru o altă partiţie extinsă. Dimensiunea acesteia corespunde exact mărimii partiţiei extinse "actuale", din care se scade domeniul de disc corespunzător unităţii logice specificate în cealaltă înregistrare a tabelei. Articolul de partiţionare extinsă a ultimei unităţi logice posedă desigur doar o singură înregistrare pentru domeniul său de pe disc şi încheie astfel încapsularea, deoarece a fost atins sfîrşitul partiţiei extinse iniţiale.

La pornirea sistemului, DOS trece prin întregul lanţ de înregistrări şi indicatoare. După care este în măsură să ataşeze fiecărei litere -unitate logică un domeniu de harddisk exact definit.

Pentru a înţelege mai bine explicaţiile de mai sus vom da un mic exemplu: un harddisk conţine 60 de cilindri şi este împărţit astfel: unitatea C ocupă 20 de cilindri, unitatea D tot 20, iar unităţile E şi F cîte 10 cilindri. În articolul de partiţionare se află la început 2 înregistrări: O partiţie DOS începînd de la cilindrul 20 pînă la 49 şi o partiţie extinsă de la cilindrul 40 pînă la 59. Articolul de partiţie extinsă conţine în primul sector al cilindrului 40 din nou două înregistrări. O partiţie DOS de la cilindrul 40 pînă la 49 şi o partiţie extinsă de la cilindrul 50 pînă la 59. În ultimul articol de partiţie extinsă în primul sector al cilindrului 50 nu se găseşte decît o singură înregistrare: o partiţie DOS de la cilindrul 50 pînă la 59. Mai rămîne doar să amintim că articolul de partiţionare este o componentă pasivă a programului FDISK.EXE şi el este copiat de acesta pe harddisk la iniţializare.

(I.M.)

DOS-extendere

Programarea sub DOS în mod protejat

În urmă cu un deceniu, primul IBM PC apărut pe piaţă era echipat cu un procesor Intel 8086 pe 16 biţi, capabil să adreseze un singur mega de memorie şi fără posibilităţi de dispecerizare a mai multor programe rezidente în memorie. Sistemul de operare i-a fost pe măsură, neprevăzînd direcţii de dezvoltare înspre exploatarea unor memorii multimegabyte partajate între mai multe aplicaţii. Zece ani mai tîrziu, MS-DOS supravieţuieşte fără schimbări majore, spre surpriza chiar a creatorilor lui, pe procesoare 80286/386, ignorînd complet modul lor protejat de adresare a memoriei şi exploatîndu-le precum microscopul-spărgător-de-nuci.

Între timp, convulsii precum standardul de memorie expandată LIM EMS 4.0 şi specificaţia de memorie extinsă XMS l-au împins pînă în faza conturării unui adevărat sistem de operare în mod protejat.

Acceptarea lentă a OS/2, inventarea extender-ului de DOS, lansarea lui Windows 3.0 şi standardizarea DPMI (interfaţa DOS - mod protejat de adresare), oferă MS-DOS-ului o nouă şansă.Nimeni nu se îndoieşte că PC-urile se îndreaptă spre un sistem de operare în mod protejat şi totuşi nu este deloc clar care va fi acesta şi cînd anume se va impune. În ciuda design-ului remarcabil şi a implementării specifice arhitecturilor 286/386, OS/2 are puţine şanse să înlocuiască DOS-ul pe scară largă. Pare cu mult mai probabil că succesorul lui MS-DOS va fi el însuşi drapat într-o formă high-tech, un nucleu de instrucţiuni pe 32 biţi deservind Windows.

În orice caz tehnologia DOS-extender "vine". Este previzibilă accentuarea competiţiei pe piaţa de soft, programele devenind tot mai complexe, pretenţiile utilizatorilor crescînd, iar programatorii vor fi nevoiţi să apeleze la extendere DOS pentru a putea îmbunătăţi performanţele şi ridica restricţiile de accesare a memoriei. Este deasemenea previzibil declinul utilizării memoriei expandate; noile implementări ale DPMI pe 386 sau procesoare mai evoluate oferă facilitatea de memorie virtuală, transformînd programarea asociată EMS într-un efort nejustificat.

Pînă în prezent extenderele DOS

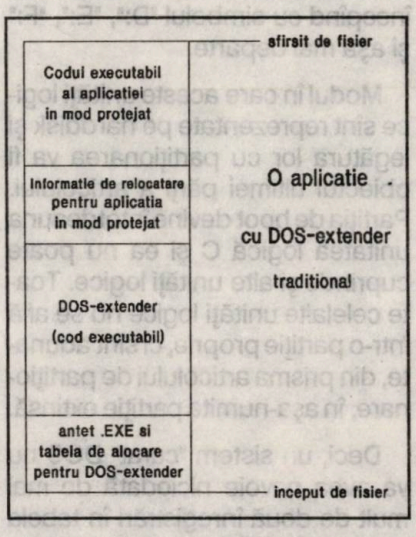


Figura 1. Fişier .EXE cu DOS-extender

au fost medii de programare relativ scumpe şi greu accesibile unui programator obişnuit. Situaţia este în schimbare rapidă, întrucît Windows 3.0 încorporează un extender DOS care în mod protejat este utilizat automat. Chiar şi aplicaţiile non-Windows pot profita de extender-ul DOS oferit de Windows 3.0, după apelul DPMI pentru comutare în mod protejat.

Practic, avantajele programării în mod protejat sînt potenţial disponibile oricărui programator care utilizează un compilator sub DOS.

Cum se utilizează un extender DOS tradiţonal ?

La procurarea unei aplicaţii anterioare lui Windows 3.0 care rulează în mod protejat, precum Mathematica, Paradox 386 sau Interleaf Publisher, se achiziţionează de fapt două programe: extender-ul DOS (provenit de la un terţ producător, precum Phar Lap sau Rational Systems) şi aplicaţia propriu-zisă.

Programatorul aplicaţiei respective trebuie să procure o licenţă de distribuţie de la producătorul extender-ului DOS pentru fiecare fişier executabil distribuit, conţinînd atît aplicaţia cît şi extender-ul propriu-zis.

Din punctul de vedere al interpretorului de comenzi MS-DOS, extender-ul DOS este singurul prezent în fişierul executabil, fiind singura parte a acestui fişier descrisă de antetul tipului EXE. MS-DOS încarcă extender-ul în memoria convenţională utilizînd modul real de adresare al procesorului, realizează toate relocările necesare şi transferă controlul extender-ului. Extender-ul DOS alocă apoi memoria adiţională (convenţională, extinsă sau amîndouă), redeschide propriul său fişier executabil şi încarcă apoi aplicaţia propriu-zisă în memorie. MS-DOS nu poate încărca direct aplicaţia în mod protejat, întrucît este absolut inocent în ceea ce priveşte selectoarele de mod protejat şi tehnicile specifice codului obiect 80386.

Extender-ul pregăteşte apoi un mediu de rulare în mod protejat a aplicaţiei: construieşte tabela de întreruperi, tabelele de descriptori locali şi globali,tabele de paginare şi segmentul de stare a programului care va furniza posibilitatea de adresare odată ce CPU este în mod protejat. Acest pas include de obicei crearea unor selectori speciali, care acoperă bufferul de împrospătare a memoriei video, prefixul segmentului de program, zona variabilelor BIOS precum şi alte structuri pe care aplicaţia ar urma să le acceseze direct. În cele

din urmă extender-ul DOS comută CPU în mod protejat şi transferă controlul adresei de început a programului.

În timp ce aplicaţia se derulează, extender-ul DOS îndeplineşte rolul său vital de interfaţă între modul protejat al aplicaţiei şi modul real DOS via controlul sistemului de întreruperi. De fapt extender-ul poate fi imaginat ca un rezident care manipulează întreruperi, care - din punctul de vedere al MS-DOS-ului - se găseşte în memoria alocată aplicaţiei şi urmează să dispară odată cu terminarea execuţiei aplicaţiei. Extender-ul DOS trebuie să facă faţă la orice întrerupere care apare în mod protejat, indiferent că este cauzată de un eveniment extern (precum ceasul de timp real), o excepţie internă (precum o împărţire la zero) sau o întrerupere software executată de aplicaţie.

În cele mai multe situaţii, cînd o întrerupere apare în mod protejat, extender-ul DOS comută pur şi simplu CPU în mod real şi transmite întreruperea rutinei de tratare iniţiale, dar totuşi unele întreruperi se bucură de un tratament special. De exemplu, Dos extender-ul de obicei reprogramează controller-ul de întreruperi 8259, realocînd noi indicative întreruperilor externe hardware de la 08h pînă la 0Fh, astfel încît aceste întreruperi pot fi diferenţiate de întreruperile interne determinate de Protecţia generală de erori, Erorile de stivă, etc. Aplicaţia poate instala propriile sale rutine de tratare în mod protejat a întreruperilor interne şi externe, apelînd funcţii speciale ale extender-ului DOS. Cele mai importante tratamente speciale apar la întreruperi utilizate pentru apeluri de funcţii: MS-DOS INT 21h, ROM BIOS, video, driver INT 10h ş.a.m.d.

Acţiunile întreprinse de extender-ul DOS, odată ce a interceptat un asemenea apel de funcţii depind de tipul de funcţie apelat. Funcţiile vizate de extender-ul DOS pot fi împărţite în patru clase de bază:

-funcţii care impun ceva mai mult decît o simplă comutare mod protejat - mod real;

-funcţii care adresează buffere ale aplicaţiei şi necesită transfer de date şi translatare de adrese;

-funcţii care trebuiesc complet înlocuite pentru a avea sens în mod protejat;

-apeluri de funcţii care sînt specifice extender-ului DOS însuşi şi furnizează servicii speciale fără echivalent în MS-DOS sau ROM-BIOS.

În prima clasă de funcţii toţi parametrii sînt transmişi în registre; parametrii nu trebuie să includă adrese. Funcţiile de intrare/ieşire pentru consolă, port serial şi imprimantă sînt exemple ilustrative. Tratarea acestor apeluri de funcţii este simplă. După interceptarea întreruperii software se salvează contextul de mod protejat, se comută CPU în mod real şi se reiniţiază întreruperea transmiţînd controlul MS-DOS-ului sau ROM-BIOS-ului. Cînd apelul de funcţie este finalizat, extender-ul DOS restaurează modul protejat (spre exemplu refac cei 16 biţi superiori ai regiştrilor de 32 de biţi ai procesorului 80386, care nu au fost neapărat menţinuţi de MS-DOS sau driverele lui) şi returnează controlul aplicaţiei.

Următoarea clasă, ceva mai complexă este compusă din funcţii ale căror parametri includ adresa unui buffer sau a altor structuri de date. În forma lor originală aceste adrese sînt lipsite de sens pentru MS-DOS sau ROM-BIOS din două motive: adresa este sub formă de selector:deplasare mai degrabă decît segment:deplasare iar buffer-ul şi datele se găsesc mai întotdeauna deasupra limitei de 1 Megabyte.

Extender-ul DOS tratează această situaţie prezervînd propriile sale buffere sub 640 K pentru o stocare temporară. Spre exemplu pentru un apel de scriere într-un fişier extender-ul DOS interceptează cererea de scriere, copiază datele din buffer în memoria convenţională, substituie adresa originală cu cea a buffer-ului de memorie convenţională, comută CPU în mod real şi în final iniţiază apelul funcţiei de scriere MS-DOS.

Cea de-a treia categorie de funcţii este cea în care extender-ul DOS trebuie să înlocuiască serviciile MS-DOS sau ROM-BIOS cu servicii noi adaptate modului protejat, înrudite cu gestiunea memoriei. Spre exemplu funcţiile MS-DOS INT 21h, 48H (alocă bloc de memorie), 49H (dealocă bloc de memorie) şi 4AH (redimensionează bloc de memorie) sînt înlocuite de funcţii DOS-extender care alocă, eliberează şi redimensionează blocuri de memorie extinsă. Funcţiile extender-ului DOS utilizează selectori de mod protejat, mai degrabă decît adrese de paragrafe sau segmente folosite de MS-DOS, dar şi substituirea este practic invizibilă unei aplicaţii " cuminţi ", exceptînd tratarea volumelor mari de memorie disponibile în modul protejat.

Ultima clasă de servicii - funcţii disponibile numai sub un extender

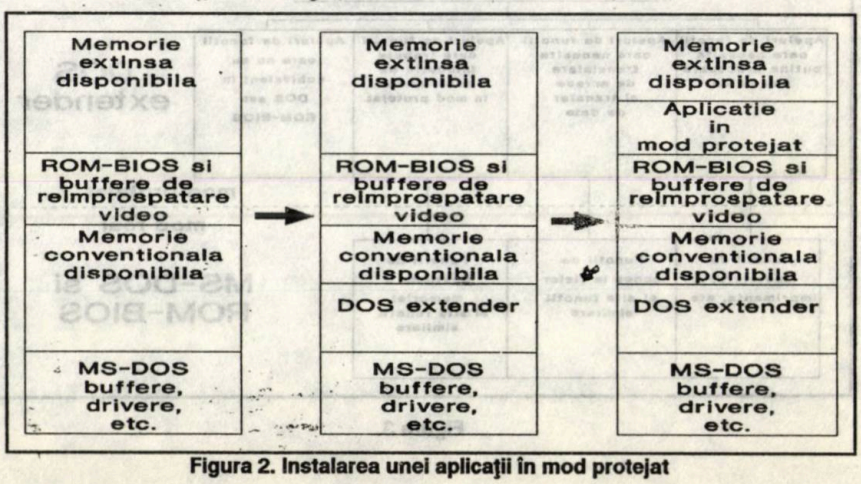


Figura 2. Instalarea unei aplicaţii în mod protejat

DOS - acoperă un spectru larg. Spre exemplu sînt apeluri de funcţii care translatează adrese din modul real în modul protejat, modifică cîmpuri de descriptori, manipulează tabela de descriptori de întreruperi, permit un apel direct al unei subrutine în mod real dintr-o aplicaţie care rulează în mod protejat, alocă blocuri de memorie convenţională, etc. Cele mai multe astfel de funcţii sînt superspecializate şi nu sînt utilizate în programe aplicative tipice.

Este de imaginat că autorii primelor DOS-extendere trebuie să fi fost virtuoşi ai programării eclectice întrucît ei realmente scriau un mic sistem de operare în mod protejat care trebuia să supravieţuiască într-un mediu foarte ostil Nu numai că aceşti creatori trebuiau să cunoască interfeţele MS-DOS şi ROM-BIOS (mai mult sau mai puţin documentate), pe dinăuntru şi pe dinafară, dar trebuiau să fie familiarizaţi cu toate trucurile neortodoxe ale unor aplicaţii indispensabile precum Lotus 1-2-3 sau Sidekick, trebuiau să stăpînească complet toate diferitele arhitecturi PC, toate caracteristicile 80286 şi 80386 şi chiar să rezolve problemele ridicate de diferite cartele acceleratoare auxiliare bazate pe 80286 şi 80386.

Pe măsură ce pachetele DPMI, altele decît Windows, devin disponibile, DOS-extenderele vor fi relativ uşor de implementat, deoarece toate rutinele dependente de hardware precum management-ul întreruperilor, comutarea modului de lucru al procesorului, întreţinerea tabelelor de pagini şi descriptori rămîn în seama interfeţei DPMI. Este probabil că în următorii cîţiva ani vom găsi funcţii de DOS-extender încorporate în bibliotecile runtime ale celor mai populare limbaje de nivel înalt, astfel încît orice program odată compilat va rula în mod protejat, utilizînd memoria extinsă dacă o interfaţă DPMI este prezentă. Deocamdată sarcina unui autor de DOS-extender este mai dură ca oricînd. Nu numai că un DOS-extender comercial, viabil trebuie să ruleze pe un sistem simplu MS-DOS, mai trebuie să coexiste paşnic cu un driver XMS de memorie extinsă, cu o interfaţă VCPI sau chiar DPMI.

Windows 3.0 DOS-extender

În Windows 3.0, un DOS-extender este încorporat în mediul de lucru spre deosebire de formula tradiţională "fiecare aplicaţie cu DOS-extender-ul propriu".

Utilizînd această abordare - de prezentare a aceleiaşi interfeţe pentru apeluri de funcţii indiferent de modul real sau protejat în care rulează programul apelant - Microsoft a evitat numeroasele probleme care au apărut sub OS/2 de la introducerea acestuia. Programatorii nu sînt confruntaţi cu o nouă API (interfaţă pentru programarea de aplicaţii), nu mai este necesară conceperea de noi documentaţii, iar bibliotecile runtime nu mai trebuiesc rescrise. Tranziţia nu este dureroasă pentru cei care au respectat regulile în elaborarea aplicaţiilor Windows.

DOS-extender-ul lui Windows este adesea confundat cu interfaţa sa DPMI - aceasta - fiind de fapt singura interfaţă DPMI diponibilă. Confuzia are o explicaţie logică perfectă. Specificaţiile interne ale lui Microsoft defineau interfaţa DPMI pe două nivele: un set de funcţii de nivel inferior pentru gestiunea întreruperilor, comutarea modurilor şi gestiunea memoriei extinse şi o interfaţă de nivel superior (DOS-extender-ul) care furnizează accesul la MS-DOS, ROM-BIOS şi driver-ul de mouse, via execuţia în mod protejat a INT 21h, INT 10h, INT 33h ş.a.m.d. Funcţiile de nivel superior erau implementate prin funcţii de nivel inferior şi serviciile modului real DOS şi ROM-BIOS.

Dacă specificaţia DPMI ar fi devenit publică într-o formă apropiată de cea originală, rezultatul imediat ar fi fost o versiune în mod protejat a MS-DOS-ului. Aceasta nu s-a întîmplat deoarece Microsoft a urmărit să elimine trăsăturile de DOS-extender ale produselor de pe piaţă pentru a impune interfaţa DPMI. Cu alte cuvinte Windows conţine totuşi propriul său DOS-extender dar facilităţile lui nu mai fac parte din specificaţia DPMI astfel încît programatorii nu mai au certitudinea prezenţei lor în interfeţe DPMI altele decît Windows 3.0. DPMI aşa cum apare astăzi este un fel de versiune din a doua generaţie a standardului precedent, VCPI (interfaţa de programare pentru control virtual), iar funcţiile sale nu se intenţionează să servească direct ca o platformă pentru o aplicaţie.

În prezent DOS-extender-ul lui Windows 3.0 este în ţara-nimănui a semidocumentaţiei şi a semisuportului - precum funcţiile TSR ale MS-DOS-ului şi interfeţele de redirectare spre reţea. Este o pilulă destul de amară.

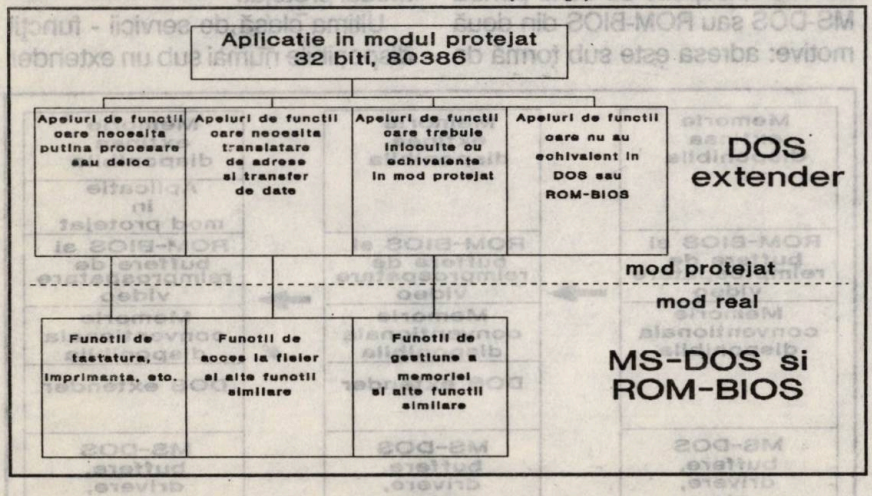


Figura 3.

Probabil că documentaţia de DOS-extender în forma originală a specificaţiei Microsoft DPMI este încă validă deoarece Windows 3.0 a fost foarte aproape de forma sa finală cînd documentaţia a fost tipărită, (din păcate sub interdicţia de publicare).

Singura documentaţie publică este o recenzie de 5 pagini apărută după un seminar Windows ISV. Acest eseu intitulat "Suportul Windows DPMI pentru INT 21h şi NETBIOS" lasă numeroase întrebări interesante fără răspuns, dar furnizează ceva informaţie critică.

Primul lucru care reiese din acest document este că deşi Windows 3.0 în mod 386 conţine întregul set de funcţii DPMI 0.9, în mod standard suportă doar un subset al funcţiilor DPMI 0.9 care permit programelor în mod protejat să comunice cu programe rezidente (TSR-uri) şi device-drivere din modul real.

Număr - Nume funcţie DPMI
0200h - Returnează vector de întrerupere în mod real
0201h - Setează vector de întrerupere în mod real
0300h - Simulează întrerupere în mod real
0302h - Apelează rutină în mod real cu IRET
0304h - Alocă adresă de întoarcere în mod real
0305h - Dealocă adresă de întoarcere în mod real.

Cea de-a doua aserţiune din acest important document este că extender-ul DOS din Windows 3.0 suportă aproape întreaga interfaţă "oficială" a MS-DOS-ului cu programele aplicative. Singurele apeluri nesuportate sînt:

INT 20h - Returnează vector de întrerupere
INT 25h - Citire absolută de pe disc
INT 26h - Scriere absolută pe disc
INT 27h - Instalare rezident

şi următoarele funcţii desuete ale INT 21h:

Funcţia INT 21h Nume MS-DOS
00h - Termină execuţia
0Fh - Deschide fişier via FCB
10h - Închide fişier via FCB
14h - Citire secvenţială cu FCB
15h - Scriere secvenţială cu FCB
16h - Crează fişier cu FCB
21h - Citire în acces direct cu FCB
22h - Scriere în acces direct cu FCB
23h - Returnează lungime fişier cu FCB
24h - Setează înregistrare relativă cu FCB
27h - Citeşte bloc cu FCB
28h - Scrie bloc cu FCB.

Doar cîteva funcţii sînt listate ca "parţial suportate" dar restricţiile sînt neesenţiale:

INT 21h, funcţiile 25h şi 35h (cere şi alocă vector întrerupere) returnează şi modifică adresele în mod protejat, neafectînd rutinele care tratează întreruperi în mod real;

Apeluri către INT 21h, funcţiile 38h şi 65h (Country Information) trebuie să ţină seama de faptul că adresa DWORD a rutinei de tratare returnată în structura de date este o adresă în mod real;

INT 21h, funcţia 44h (IOCTL), subfuncţiile 2,3,4 şi 5 restrîng dimensiunea datelor la mai puţin de 4K dacă buffer-ul este deasupra limitei de 1MB în timp ce facilităţile de coduri de pagină ale IOCTL, subfuncţia 0Ch nu mai sînt suportate deloc.

Ce noutăţi despre apelurile de ROMBIOS? Specificaţia originală Microsoft DPMI listează în mod explicit practic fiecare apel ROMBIOS şi dacă este sau nu suportat. Documentul public de cinci pagini afirmă numai: "Dacă o întrerupere software API este în întregime bazată pe registre, fără pointeri, registre de segment şi parametri de stivă, acea API ar trebui să ruleze sub Windows în mod protejat". Din fericire, aproape toate funcţiile de driver Microsoft Mouse (INT 33h) şi ROMBIOS video (INT 10h), comunicaţii (INT 14h), tastatură (INT 16h), imprimantă (INT 17h), dată şi timp (INT 1Ah) se plasează sub umbrela aplicaţiilor bazate pe registre şi sînt de aşteptat să funcţioneze corespunzător.

Programarea într-un limbaj de nivel înalt în mod protejat sub Windows este uşoară odată scrisă şi depanată o unică rutină care comută în mod protejat printr-un apel la interfaţa DPMI. Cu un pic de noroc biblioteca runtime a compilatorului nu va tolera următoarele practici care crează probleme în mod protejat:

-utilizarea registrelor de segment pentru stocare;

-operaţiuni aritmetice cu conţinutul registrelor de segment;

-comparare directă a doi pointeri din segmente diferite;

-citire/scriere în afara limitelor memoriei alocate;

-cod care se automodifică.

(A.F.)

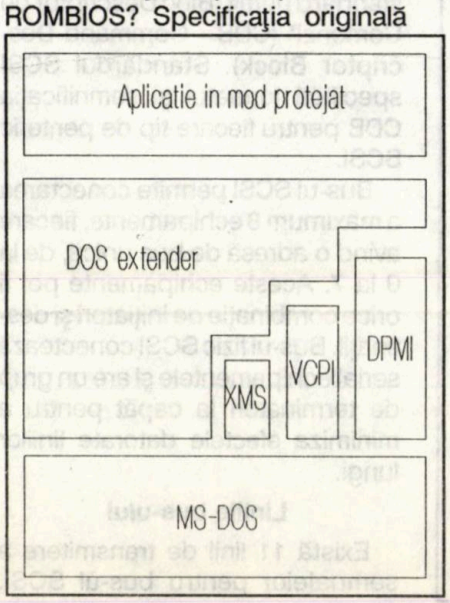


Figura 4

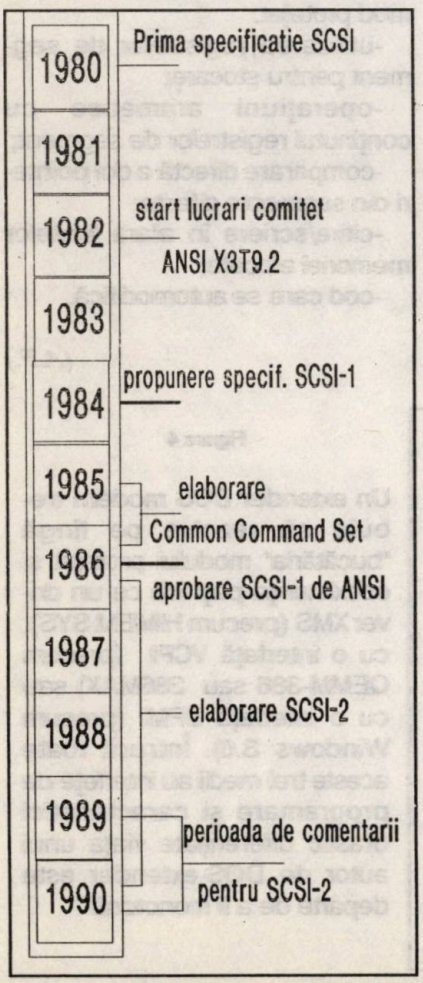
Un extender DOS modern trebuie să rezolve pe lîngă "bucătăria" modului protejat şi coexistenţa paşnică cu un driver XMS (precum HIMEM.SYS), cu o interfaţă VCPI (precum QEMM-386 sau 386MAX) sau cu o interfaţă DPMI (precum Windows 3.0). Întrucît toate aceste trei medii au interfeţe de programare şi caracteristici drastic diferenţiate viaţa unui autor de DOS-extender este departe de a fi monotonă !

Interfaţa SCSI - Prezentare şi utilizare

Privire retrospectivă

Interfaţa SCSI (Small Computer System Interface, pronunţat în limba engleză şi ca "scuzzy") lucrează la nivelul sistemului şi a fost elaborată pentru a permite comunicaţiile între sistemele de calcul mici şi perifericele lor. Prima versiune a fost realizată la Shugart Associates, în versiunea SASI (Shugart Associates System Interface) pentru legătura între controllerul de disc şi discurile hard produse de firmă. În 1979 ANSI (American National Standards Institute) a adoptat şi standardizat acest sistem de interfaţă tip bus, l-a redenumit SCSI şi i-a adăugat facilităţi. SCSI fiind o specificaţie de interfaţă la nivel sistem, defineşte nu numai nivelele semnalelor pe bus şi funcţia lor logică, ci specifică şi protocolul de comunicaţie şi secvenţe de comenzi. Aceasta duce la o mare autonomie a echipamentului conectat pe bus. Controller-ul de pe bus-ul SCSI poate comunica cu discuri floppy, discuri hard, streamere, etc. fără a ţine seama de diferenţele tehnice interne de realizare a acestor periferice. Se pot realiza modernizări în tehnologia de producţie a echipamentelor compatibile SCSI fără schimbări majore în soft-ul şi hard-ul de interfaţă, acestea păstrîndu-şi modul de funcţionare standard.

(R)Evoluţia SCSI



Structura generală

Comunicarea pe bus-ul SCSI are loc între un echipament care iniţiază transferul (initiator) şi un echipament destinaţie (target). Comunicarea pe bus-ul SCSI este permisă doar între două echipamente, la un moment dat. Unul dintre echipamente lucrează ca iniţiator, selectînd şi comandînd echipamentul destinaţie care efectuează operaţia dorită. Cînd comunicaţia este stabilită între iniţiator şi destinaţie, comenzile spre destinaţie sînt emise într-un format standard numit "Bloc Descriptor de Comenzi" (CDB - Command Descriptor Block). Standardul SCSI specifică formatul şi semnificaţia CDB pentru fiecare tip de periferic SCSI.

Bus-ul SCSI permite conectarea a maximum 8 echipamente, fiecare avînd o adresă de bus unică, de la 0 la 7. Aceste echipamente pot fi orice combinaţie de iniţiatori şi destinaţii. Bus-ul fizic SCSI conectează serial echipamentele şi are un grup de terminatori la capăt pentru a minimiza efectele datorate liniilor lungi.

Liniile bus-ului

Există 11 linii de transmitere a semnalelor pentru bus-ul SCSI. Nouă dintre acestea sînt folosite pentru date (8 biţi date plus un bit de paritate impară) şi nouă pentru control. Semnalele sînt:

BSY (Busy) o linie realizată SAU cablat, care indică faptul că bus-ul este ocupat.

SEL (Select) este folosit de iniţiator pentru selectarea destinaţiei sau de destinaţie pentru a reselecta un iniţiator.

C/D (Command/Data) comandat de destinaţie pentru a indica dacă se vor transmite informaţii de date sau control pe bus.

I/O (Input/Output) comandat de destinaţie pentru a controla direcţia de transfer a datelor pe bus.

MSG (Message) comandat de destinaţie în timpul fazei de mesaj.

REQ (Request) comandat de destinaţie pentru a indica o cerere de dialog pentru transfer de date.

ACK (Acknowledge) comandat de iniţiator pentru a indica recunoaşterea într-un dialog REQ/ACK (hand shake)

ATN (Attention) comandat de iniţiator pentru a indica o condiţie de atenţie pentru destinaţie.

RST (Reset) o linie SAU cablată care iniţializează (reset) bus-ul SCSI şi resetează toate echipamentele de pe bus.

DB 0-7, P (Data bus) opt linii de date şi o linie de paritate asociată lor. Transferul tuturor datelor şi comenzilor are loc pe liniile de date. Semnificaţia exactă a octeţilor de date şi direcţia transferului sînt determinate de destinaţie, prin liniile de comandă C/D, I/O şi MSG.

Liniile de dialog (Handshake lines)

Nivelul cel mai de jos al protocolului de transfer SCSI este prin "handshake". Pentru fiecare octet transmis în oricare direcţie, are loc un dialog pe liniile de cerere (REQ) şi recunoaştere (ACK) cerere. Dialogul SCSI este total asincron.

Procesul de dialog începe cînd destinaţia stabileşte cererea (REQ). Cînd iniţiatorul sesizează REQ, răspunde citind sau scriind un octet de pe/pe bus-ul de date şi poziţionînd linia ACK. Cînd destinaţia sesizează ACK răspunde prin aducerea în stare iniţială a lui REQ. Acesta duce la aducerea în stare iniţială a liniei ACK. Bus-ul este pregătit pentru o nouă secvenţă de dialog REQ;ACK. Acest proces de dialog este foarte fiabil, deoarece este nevoie de verificarea stării semnalului la fiecare capăt înaintea iniţierii unui nou dialog.

Faze

Secvenţele de comunicaţii pe bus-ul SCSI trec prin mai multe faze distincte (BUS FREE, ARBITRATION, SELECTION, COMMAND, DATA, STATUS, MESSAGE, RESELECTION). Bus-ul nu poate fi niciodată în mai mult de o singură fază la un moment dat. După selectare, schimbările de fază sînt controlate de destinaţie prin liniile de bus C/D, I/O şi MSG.

Faza BUS FREE (Bus liber)

În această fază nu există echipamente SCSI care să folosească bus-ul şi acesta este disponibil. Toate comunicaţiile SCSI încep din faza BUS FREE.

Faza ARBITRATION (de arbitrare)

Această fază are loc doar la sisteme SCSI care au mai mult decît un singur iniţiator sau o destinaţie cu reselectare. De vreme ce este permisă existenţa pe bus a mai multor iniţiatori, pot exista cazuri cînd aceştia încearcă să acceseze bus-ul simultan. Pentru a putea arbitra conflictele se ataşează priorităţi adreselor echipamentelor SCSI de pe bus. Prioritatea maximă o are adresa 7, cea minimă adresa 0.

Fiecărei adrese SCSI i se atribuie un bit de date, pentru a fi folosit în timpul fazei de arbitrare (de ex. DB0 este folosit de adresa 0 şi DB7 de adresa 7). În timpul fazei de arbitrare, fiecare echipament SCSI care doreşte să ocupe bus-ul poziţionează BSY, un semnal SAU cablat, şi de asemenea lotul de date corespunzător adresei de pe bus. Dacă după două microsecunde, există biţi poziţionaţi cu prioritate mai mare, echipamentul trebuie să elibereze bus-ul şi să aştepte o altă fază BUS FREE.

Faza SELECTION (de selectare)

După ce a avut cîştig de cauză în faza ARBITRATION iniţiatorul trece în faza de selectare SELECTION. În această fază poziţionează semnalul SEL şi biţii de date corespunzători atît adresei proprii cît şi cel a destinaţiei care se va selecta. Astfel destinaţia poate determina ce adresă SCSI are iniţiatorul.

Faza COMMAND (de comenzi)

În această fază destinaţia cere informaţie de comenzi de la iniţiator. Informaţia de comenzi este sub forma de Bloc descriptor de comenzi CDB (Command Descriptor Block). Acesta conţine informaţia necesară pentru execuţia oricărei operaţii definite pentru o clasă anume de echipament SCSI, în format standard.

Faza DATA (de date)

În această fază destinaţia transferă toate datele cerute de Blocul descriptor de comenzi CDB încărcat în faza anterioară. Unele comenzi SCSI nu necesită transfer de date sau vor sări peste această fază dacă se detectează eroare.

Faza STATUS (stări)

În această fază se trimit informaţii de stare despre ultimul transfer către iniţiator. Faza STATUS urmează imediat după DATA, sau dacă această fază nu a fost necesară, după COMMAND. Starea returnată are lungime de 1 byte. Biţii din acest octet sînt cu semnificaţie predefinită de specificaţiile SCSI. Există trei biţi liberi pentru implementări specifice furnizor.

Organizarea STATUS BYTE

Starea "ok" indică terminarea cu succes a operaţiei precedente. Condiţia de "check" ne semnalează că s-a întîmplat ceva neprevăzut în cursul operaţiei SCSI. Iniţiatorul trebuie să trimită o comandă de "REQUEST SENSE" (cerere sesizare) pentru a localiza evenimentul neprevăzut. Busy indică faptul că destinaţia nu poate accepta comandă şi iniţiatorul trebuie să încerce din nou.

	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	ex.
	r	f	f	s	s	s	s	f	
ex.	r	f	f	0	0	0	0	f	ok
	r	f	f	0	o	0	1	f	check
	r	f	f	0	1	0	0	f	busy

Status byte

Semnificaţie: r=rezervat, s = predefinit SCSI, f = specific furnizor

Faza MESSAGEIN (mesaj la iniţiator)

Aceasta este ultima fază în cadrul unui transfer SCSI. Se transferă un octet de mesaj de la destinaţie la iniţiator. După transferul octetului echipamentul care era destinaţie se deconectează de la bus-ul SCSI şi revine în faza BUS FREE. în acest moment bus-ul SCSI devine pregătit pentru o altă operaţie de transfer SCSI.

Faza MESSAGEOUT (mesaj la destinaţie)

După terminarea fazei SELECTION de către iniţiator, destinaţia are controlul complet al transferurilor de date şi schimbării fazelor.

Dacă iniţiatorul are un motiv de recîştigare a controlului pe bus (de exemplu detectarea unei erori de paritate) singura lui posibilitate este poziţionarea liniei ATN. Dacă destinaţia detectează poziţionarea liniei ATN în timpul unei operaţii SCSI, atunci va trece în faza de MESSAGE OUT. În acest caz iniţiatorul poate trimite un octet de mesaj (dacă ATN este menţinut se pot trimite mai mulţi octeţi) pentru a forţa destinaţia să abandoneze şi să reîncerce operaţia SCSI. Specificaţia SCSI defineşte multe configuraţii de octeţi de mesaj, obligatorie fiind în orice implementare mesajul OO indicînd terminarea comenzii (Command Complete).

Reconectarea

Pentru a preveni ocuparea pentru perioade lungi în cazul transferurilor lente, anumite periferice de pe bus-ul SCSI au facilitatea numită "reconectare". La operaţii lente, ca de exemplu căutarea (seek) îndelungată pe disc, destinaţia va accepta un bloc descriptor de comenzi CDB descriind operaţia, apoi se va deconecta în mod deliberat de pe bus.

Cînd destinaţia devine gata (ready) pentru faza următoare a transferului, va acţiona ca iniţiator pentru a selecta iniţiatorul operaţiei şi apoi îşi va relua rolul de destinaţie. Destinaţia îşi va relua acum rolul în faza următoare pentru a completa operaţia.

Blocul descriptor de comandă CDB (Command Descriptor Block)

Blocurile predefinite descriptoare de comenzi sînt o parte importantă din ceea ce face ca bus-ul SCSI să fie performant şi popular. Folosind un set obligatoriu de comenzi, se pot scrie drivere soft pentru orice echipament SCSI. Acesta este o parte din ceea ce diferenţiază SCSI ca interfaţă sistem, de interfeţele la nivel echipament. SCSI specifică nu numai nivelele de semnal şi temporizările pentru interfaţă ci şi comenzi de nivel înalt, la care trebuie să se conformeze orice echipament pentru a fi un echipament SCSI.

Formatul exact şi semnificaţia octeţilor din CDB depind de tipul de echipament. În mod obişnuit se descriu 6 tipuri de echipamente, împreună cu un set minim de comenzi obligatorii.

Echipament . Exemplu

cu acces direct - disc floppy, harddisc
acces secvenţial - unitate de bandă, streamer,
imprimantă - matricială, laser
procesor - orice echipament inteligent
WORM - (write once, read multiple) medii optice
acces direct, citire - CD ROM

Blocurile CDB pot avea trei dimensiuni: 6 octeţi, 10 octeţi şi 12 octeţi. Primul octet la orice CDB este un cod operaţie, şi o parte din acesta specifică lungimea CDB. Octetul cod operaţie este divizat astfel: biţii 7-5 conţin codul de grup iar biţii 4-0 codul comenzii. Biţii cei mai semnificativi sînt deci folosiţi pentru codul de grup. În prezent avem 3 coduri de grup definite: grupurile 0,1 şi 5.

CDB-urile din grupul 0 au totdeauna lungimea de 6 octeţi, din grupul 1 de 10 octeţi iar cele din grupul 5 au lungimea de 12 octeţi. Grupurile 2 şi 4 sînt rezervate pentru dezvoltări ulterioare iar 6 şi 7 pot fi utilizate pentru implementări specifice furnizor.

Cîmpul de 5 biţi de cod de comandă furnizează pînă la 32 comenzi pentru fiecare grup. Formatele pentru CDB-urile de 6 şi 10 octeţi sînt date în figură:

grupa 0:

octet
0 cod operaţie
1 LUN(db7-5) / nr. bloc de start (dacă există) MSB
2 nr. bloc de start (dacă există)
3 nr. bloc de start (dacă există) LSB

Semnal	Pin	Pin	Semnal
masă ecran	1	2	masă
+db0	3	4	-db0
+db1	5	6	-db1
+db2	7	8	-db2
+db3	9	10	-db3
+db4	11	12	-db4
+db5	13	14	-db5
+db6	15	16	-db6
+db7	17	18	-db7
+dbp	19	20	-dbp
diffsens	21	22	masă
masă	23	24	masă
termpwr	25	26	termpwr
masă	27	28	masă
+atn	29	30	-atn
masă	31	32	masă
+bsy	33	34	-bsy
+ack	35	36	-ack
+rst	37	38	-rst
+msg	39	40	-msg
+sel	41	42	-sel
+c/d	43	44	-c/d
+req	45	46	-req
+i/o	47	48	-i/o
masă	49	50	masă

Asignarea la pini pentru Differential-SCSI

Pin	Semnal
2	-db0
4	-db1
6	-db2
8	-db3
10	-db4
12	-db5
14	-db6
16	-db7
18	-dbp
20	masă
22	masă
24	masă
26	termpwr
28	masă
30	masă
32	-atn
34	masă
36	-bsy
38	-ack
40	-rst
42	-msg
44	-sel
46	-c/d
48	-req
50	-i/o

Single-ended SCSI

Toţi pinii impari sînt legaţi la masă

4 lungime transfer (dacă există)
5 octet de control
grupa 1
0 cod operaţie
1 LUN (db7-5)/rezervat(db4-1) / rel (db0)
2 nr. bloc de start (dacă există) MSB
3 nr. bloc de start (dacă există)
4 nr. bloc de start (dacă există)
5 nr. bloc de start (dacă există) LSB
6 rezervat
7 lungime transfer (dacă există) MSB
8 lungime transfer (dacă există) LSB
9 octet de control

Bus-ul SCSI există în două variante: single-ended-SCSI, la care fiecare semnal e transmis ca tensiune faţă de o masă comună, şi differential SCSI, la care semnalul este reprezentat de diferenţa de tensiune între două conductoare pereche. Specificaţia SCSI limitează lungimea unui single-ended-bus la maxim 6 metri, avînd în vedere mai ales conexiuni în interiorul unei carcase. Un differential-bus în schimb poate fi lung de pînă la 25 metri, fiind mai indicat din acest punct de vedere pentru conectarea unor echipamente aflate în amplasamente distincte. Din motive de siguranţă, echipamente conectate prin cele două metode nu ar trebui amestecate pe acelaşi bus.

SCSI-1 sau SCSI-2 ?

Dacă SCSI-1 deja s-a afirmat, SCSI-2 de-abia "se naşte". Care e diferenţa între ele? Citînd un membru al comitetului ANSI pentru SCSI: 'SCSI promite flexibilitate şi performanţe superioare, iar prin SCSI-2 acestea se obţin mai simplu'. SCSI-1 este un standard definit (ANSI X3.131-1986), iar SCSI-2 încă se discută. (A nu se confunda cele două standarde cu cele două variante, single-ended şi differential, pomenite mai sus - acestea apar ambele atît în SCSI-1 cît şi în SCSI-2 !)

A existat o vreme în care se părea că ESDI răspunde tuturor cerinţelor utilizatorilor, mai ales controller-ele noi Super-ESDI cu rate de transfer de pînă la 24 Mbiţi/secundă. Dar SCSI este standardul necontestat în lumea staţiilor de lucru Unix şi a Macintosh-urilor şi se bucură de o atenţie din ce în ce mai mare şi în lumea PC-urilor. Există motive tehnice clare în favoarea acestei opţiuni; totuşi, utilizatorii trebuie să fie circumspecţi. Căci în mediul PC-urilor există diverse piedici în calea implementării subsistemelor SCSI şi sînt unele probleme de compatibilitate şi performanţă. Un avantaj al SCSI este viteza sa teoretică. Deja primele specificaţii SCSI permit rate de transfer de 4 Mocteţi/s (32 Mbiţi/s), ceea ce este mai mult decît oferă Super-ESDI. Cu toate acestea, deseori SCSI este "simţit" ca fiind mai încet. Acest lucru se datorează implementărilor actuale pentru PC-uri. Spre deosebire de ESDI, IDE sau ST-506, SCSI este un standard de bus, nu de echipament. În felul acesta SCSI oferă o interfaţă logică între calculator şi periferia conectată, care face calculatorul independent de caracteristicile hard ale acesteia. Astfel, teoretic, orice harddisk SCSI poate fi conectat la un host-adapter, fără nici o grijă relativ la BIOS sau firmware-ul din controller. La fel de simplu se pot conecta alte echipamente SCSI: scannere, streamere, imprimante etc. În staţiile de lucru sau Macintosh-uri se poate astfel integra în sistem o paletă extrem de bogată de echipamente periferice variate, fără a ridica măcar capacul de pe carcasa calculatorului. Echipamentele se conectează pur şi simplu printr-un cablu SCSI şi gata, pot fi folosite la întreaga capacitate de către sistem, fără nici un efort suplimentar. Această flexibilitate se numără printre avantajele majore oferite de SCSI.

Şi totuşi, pe măsură ce tot mai mulţi producători se orientează către SCSI, distanţa dintre teorie şi realitate creşte. Multe din echipamentele SCSI ale trecutului erau, în realitate, incompatibile.

O altă problemă este "overhead"-ul condiţionat de SCSI. Interfaţa logică are nevoie de timp de calcul, pentru a transforma comenzile logice în succesiunea corespunzătoare de comenzi fizice. Iar acest overhead se repetă la fiecare comandă SCSI.

La teste efectuate mai demult, PC-uri cu sisteme SCSI s-au dovedit a lucra de-abia la jumătatea vitezei celor de tip ESDI sau ST-506! Dar în prezent, piaţa oferă echipamente SCSI (host-adaptere şi echipamente periferice) pentru toate variantele. Software şi drivere sînt disponibile pentru DOS, OS/2 şi Unix, harddisk-urile SCSI sînt la fel de rapide ca cele ESDI. Iar faptul că şi IBM şi DEC s-au orientat spre SCSI pare să asigure semnificaţia SCSI ca "standard al anilor '90". Marea majoritate a echipamentelor periferice disponibile corespund standardului SCSI-1. Dar produse de performanţe ridicate, comparabile cu perifericele mainframe-urilor, au început deja să apară, bazate pe SCSI-2. Caracteristicile cele mai importante ce definesc SCSI-2:

- rate de transfer de pînă la 10 Mocteţi/s prin busul SCSI de 8 biţi
- specificaţii pentru bus de date de 16 şi respectiv 32 biţi, cu rate de transfer posibile de pînă la 40 Mocteţi/s
- corecţia de erori la nivelul echipamentelor, pentru a păstra disponibilitatea unităţii centrale pentru alte activităţi
- cozi de aşteptare pentru comenzi care permit echipamentelor SCSI să-şi optimizeze singure performanţele
- nu în ultimul rînd: compatibilitate perfectă cu SCSI-1

Chiar dacă încă mai sînt probleme în atingerea vitezelor maximale de transfer permise de SCSI-2, toate elementele necesare pentru a face posibile aceste viteze de lucru sînt date. Ceea ce înseamnă că era SCSI a început.

ing. Attila Darvas, ing. Iosif Fettich

"Lupa de timp"

Deseori, ar putea fi utilă reducerea temporară a vitezei de execuţie a PC-ului, pentru a urmări mai bine o succesiune de imagini grafice de exemplu. Programul "LUPA" permite "frînarea" PC-ului, cu ajutorul combinaţiei de taste <Ctrl>+<Alt>+<Space>. Frîna este activă atît timp cît se ţine apăsată combinaţia de taste menţionată. Temporizarea se acordează individual; valorile ce determină întîrzierea sînt cele din rîndul 34 şi respectiv 36 (cu cît produsul AX * CX este mai mare, cu atît întîrzierea produsă este mai mare). Programul se creează cu TASM: TASM LUPA; TLINK LUPA; EXE2BIN LUPA.EXE LUPA.COM. Se poate intra direct şi în DEBUG; (vezi programul din coloana alăturată); pentru a-i ajuta pe cei care preferă această variantă (sau nu au un macroasamblor disponibil) am dat şi valorile corespunzătoare ale adreselor ca şi comentarii.

(C.R.)

```
1  o|  title    #lupa.asm                                    |o
2  o|  extrn    afara: far                                   |o
3  o|  code     segment word                                 |o
4  o|  assume   cs:code, ds:code                             |o
5  o|           org      100h                                |o
6  o|  start:                                                 |o
7  o|           mov      ax, 3509h                           |o
8  o|           int      21h          ;103                   |o
9  o|           push     es           ;105                   |o
10 o|           push     bx           ;106                   |o
11 o|           pop      [word ptr x+1]  ;[144]  ;107        |o
12 o|           pop      [word ptr x+3]  ;[146]  ;10b        |o
13 o|           mov      dx, offset newint9 ;11e  10f        |o
14 o|           mov      ax, 2509h    ;112                   |o
15 o|           int      21h          ;115                   |o
16 o|           mov      dx, 15h      ;117                   |o
17 o|           mov      ah, 31h      ;11a                   |o
18 o|           int      21h          ;11c                   |o
19 o|  ; noul  int 9h                                         |o
20 o|  newint9:push      ax           ;11e                   |o
21 o|           push     ds           ;11f                   |o
22 o|           push     cx           ;120                   |o
23 o|  ; test  hotkey                                         |o
24 o|           in       al, 60h      ;121                   |o
25 o|           cmp      al, 39h      ;123                   |o
26 o|           jnz      exit         ;142   ;125            |o
27 o|           xor      ax, ax       ;127                   |o
28 o|           mov      ds, ax       ;129                   |o
29 o|           mov      al, ds:[417h]  ;12b                 |o
30 o|           and      al, 0fh      ;12e                   |o
31 o|           cmp      al, 0ch      ;130                   |o
32 o|           jnz      exit         ;142   ;132            |o
33 o|  ; inceput bucla de intirziere                          |o
34 o|           mov      ax, 4h       ;134                   |o
35 o|  again:   dec      ax           ;137                   |o
36 o|           mov      cx, 2000h    ;138                   |o
37 o|  bucla:   loop     bucla        ;13b   ;13b            |o
38 o|           cmp      ax, 0        ;13d                   |o
39 o|           jnz      again        ;137   ;140            |o
40 o|  ; sfirsit bucla                                        |o
41 o|  ;revenire in vechiul int9                              |o
42 o|  exit:    pushf                 ;142                   |o
43 o|  x:       call     far ptr afara  ;143                 |o
44 o|           pop      cx           ;148                   |o
45 o|           pop      ds           ;149                   |o
46 o|           pop      ax           ;14a                   |o
47 o|           iret                  ;14b                   |o
48 o|  code     ends                                         |o
49 o|  end      start                                        |o
```

LUPA.ASM

Tabela ASCII confortabilă

Programul de mai jos, pe care îl puteţi folosi ca atare ca intrare în SID (DR-DOS; SID < ASCII.DEB) afişează pe ecran o tabelă ASCII; Dacă în linia de comandă mai specificaţi caractere suplimentare, acestea apar în tabelă în mod blinking şi în altă culoare. Pentru intrare în DEBUG, liniile 54-55 se înlocuiesc cu secvenţa:

54 rcx 55 74 56 nasc2.com 57 w 58 q

(C.R.)

```
1  o|  a100                                                 |o
2  o|  mov    ah,f                                          |o
3  o|  int    10                                            |o
4  o|  mov    si,ff00                                       |o
5  o|  mov    ax,si                                         |o
6  o|  call   147                                           |o
7  o|  mov    al,1a                                         |o
8  o|  call   160                                           |o
9  o|  mov    ax,si                                         |o
10 o|  mov    bl,b                                          |o
11 o|  mov    di,80                                         |o
12 o|  inc    di                                            |o
13 o|  cmp    by[di],0                                      |o
14 o|  jz     124                                           |o
15 o|  cmp    [di],al                                       |o
16 o|  jnz    118                                           |o
17 o|  mov    bl,8c                                         |o
18 o|  call   162                                           |o
19 o|  mov    al,20                                         |o
20 o|  call   160                                           |o
21 o|  add    si,14                                         |o
22 o|  jnb    107                                           |o
23 o|  mov    ax,e0d                                        |o
24 o|  int    10                                            |o
25 o|  mov    ax,e0a                                        |o
26 o|  int    10                                            |o
27 o|  inc    by[105]                                       |o
28 o|  cmp    by[105],14                                    |o
29 o|  jb     104                                           |o
30 o|  ret                                                  |o
31 o|  mov    cx,3003                                       |o
32 o|  aam                                                  |o
33 o|  add    al,ch                                         |o
34 o|  push   ax                                           |o
35 o|  cmp    ah,0                                          |o
36 o|  jnz    156                                           |o
37 o|  mov    ch,20                                         |o
38 o|  dec    cl                                            |o
39 o|  jz     15f                                           |o
40 o|  mov    al,ah                                         |o
41 o|  call   14a                                           |o
42 o|  pop    ax                                            |o
43 o|  mov    bl,7                                          |o
44 o|  mov    cx,1                                          |o
45 o|  mov    ah,9                                          |o
46 o|  int    10                                            |o
47 o|  mov    ah,3                                          |o
48 o|  int    10                                            |o
49 o|  inc    dl                                            |o
50 o|  mov    ah,2                                          |o
51 o|  int    10                                            |o
52 o|  ret                                                  |o
53 o|                                                       |o
54 o|  wasc2.com,100,174                                    |o
55 o|  q                                                    |o
```

ASCII.DEB

Tabelă ASCII rezidentă

Programul ASCII.PAS creează un program rezident (TSR) care intră în acţiune pe "hotkey" (combinaţia de taste) <Ctrl> + <Alt> + <A>. Cît timp ţineţi apăsate cele trei taste simultan, pe ecran vă apare o tabelă Ascii extinsă (256 de caractere). Cînd eliberaţi tastele, tabela dispare şi puteţi continua lucrul din punctul în care l-aţi întrerupt Funcţie de cartela grafică existentă în sistemul Dvs.., s-ar putea să trebuiască să modificaţi adresa memoriei ecran (linia 5)

(C.R./I.F.)

Economie de memorie RAM

Deseori, datorită multiplelor programe rezidente încărcate, memoria RAM a PC-ului se dovedeşte neîncăpătoare. Nu se prea ştie însă că, în anumite limite, spaţiul folosit de acestea poate fi optimizat Faptul că sistemul construieşte şirul "environment" al unui program folosind toţi parametri de "environment ai programului apelant (de regulă, COMMAND.COM), duce la concluzia că procedura de urmat este ca întîi să se instaleze toate TSR-urile, iar abia apoi să se fixeze parametri "environment" prin intermediul comenzilor SET, PATH, DOS. Procedînd în felul acesta, se pot economisi pînă în 100 octeţi per program. O altă posibilitate este folosirea DEVICE-driverelor în locul unor programe ce se autoinstalează la apel. Pentru mouse de ex., de regulă driverul este furnizat în două variante: ca program executabil (MOUSE.COM) sau ca device-driver (MOUSE.SYS). Înglobarea în sistem via CONFIG.SYS duce la o economie de 200 octeţi faţă de instalarea prin apel DOS. Diferenţa provine din faptul că la instalarea unui program TSR, DOS mai foloseşte memorie, în afara codului efectiv, pentru şirul "environment", segmentul ce prefixează un program, blocurile de control a memoriei - toate acestea nu sînt necesare cînd instalarea se face prin CONFIG.SYS.

Cîtă memorie folosesc programele pe care le aveţi încărcate la un moment dat, puteţi urmări , de ex., cu comanda MEM /DEBUG (DOS 4.0)

(C.R./L.W.)

```
1   o|   {$M 1024,,0,,0}                                                                  |o
2   o|   uses Dos;                                                                        |o
3   o|   type ecran = array[0..1999] of word;                                             |o
4   o|   var shift : byte absolute $40:$17;                                               |o
5   o|           SA   : ecran absolute $b800:0; {eventual $b000:0, adresa memoriei ecran}  |o
6   o|           SB   : ecran;                                                            |o
7   o|           i, p, t: word;                                                           |o
8   o|   const H   : String = 'Tabela ASCII:';                                            |o
9   o|           flag : boolean = false;                                                  |o
10  o|                                                                                     |o
11  o|   procedure Int9 (Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);        |o
12  o|   Interrupt;                                                                        |o
13  o|   begin                                                                            |o
14  o|           if flag then begin                                                       |o
15  o|                   if (shift and 12 = 12) and (port[$60] = 158) then begin          |o
16  o|                           flag := false;                                           |o
17  o|                           SA := SB;                                                |o
18  o|                   end;                                                             |o
19  o|           end else begin                                                           |o
20  o|           if (shift and 12 = 12) and (port[$60] = 30) then begin                   |o
21  o|                           flag := true;                                            |o
22  o|                           SB := SA;                                                |o
23  o|                           for i := 0 to 1999 do SA[i] := 0;                        |o
24  o|                           for i := 1 to ord(h[0]) do                               |o
25  o|                                   SA[i] := ord (h[i]) + 1792;                      |o
26  o|                           for i := 0 to 255 do begin                               |o
27  o|                                   p := (i mod 24*80 + 80) + (i div 24*7 +2);       |o
28  o|                                   SA[p] := i+1792;                                 |o
29  o|                                   SA[p+2] := i div 100 + 1840;                     |o
30  o|                                   SA[p+3] := i mod 100 div 10 + 1840;              |o
31  o|                                   SA[p+4] := i mod 10 + 1840;                      |o
32  o|                           end;                                                     |o
33  o|                   end;                                                             |o
34  o|           end;                                                                     |o
35  o|           Inline ($9c/$9a/0/0/0/0);                                                |o
36  o|   end;                                                                             |o
37  o|                                                                                     |o
38  o|   begin                                                                            |o
39  o|           WriteLn ('Tabela ASCII rezidenta');                                      |o
40  o|           WriteLn ('Se activeaza prin Ctrl-Alt-A');                                |o
41  o|           MemW[Seg(Int9):Ofs(Int9)+$1a4] := MemW [0:36];                           |o
42  o|           {pardon...Atentie daca modificati vreo linie de cod!                     |o
43  o|                   mai treceti odata prin debug, sa vedeti unde pica...}            |o
44  o|           MemW[Seg(Int9):Ofs(Int9)+$1a6] := MemW [0:38];                           |o
45  o|           SetIntVec($9, @Int9); Keep (0);                                          |o
46  o|   end.                                                                             |o
```

ASCII.PAS

Igiena zilnică

Pentru întreţinerea harddisk-ului este necesar ca periodic să se efectueze anumite operaţii, ca de ex.:

- căutarea sectoarelor defecte;
- căutarea programelor infestate de viruşi;
- salvarea informaţiilor vitale (sectorul de boot, FAT-ul şi structura de directoare) ;
- reformatarea nedistructivă

În prezent există o mulţime de produse-program care execută unul sau mai multe din sarcinile de mai sus. Execuţia lor la comanda operatorului are dezavantajul că nu asigură ritmicitate. Pe de altă parte, execuţia automată a acestor programe prin înscrierea lor în fişierul AUTOEXEC.BAT nu este recomandată datorită timpului consumat la iniţializare.

Programul propus asigură execuţia automată a acestor date, la anumite intervale de timp.

Utilitarele sînt împărţite în 3 grupe:

- utilitare ce se execută zilnic, la prima pornire a calculatorului;
- utilitare ce se execută săptămînal (în fiecare miercuri);
- utilitare ce se execută lunar (în prima zi a lunii);

Utilitarele din cele 3 categorii sînt cuprinse în fişierele EVERYDAY.BAT, WEEKLY.BAT şi MONTHLY.BAT, ceea ce permite schimbarea lor fără recompilarea programului.

În fişierul AUTOEXEC.BAT se va insera o linie cuprinzînd apelul programului:

C:\DAILY

Pentru calculatoarele compatibile XT, care nu au incorporat in ceas intern, această linie trebuie să se găsească după comenzile de introducere de la tastatură a datei şi orei (DATE, respectiv TIME).

Alăturat sînt prezentate listingurile programului şi exemple de programe ce pot să apară în cele 3 fişiere de comenzi.

Brînduş Vasile,
str. Aurora nr. 25, bl. 10, apt. 32, Braşov

```
program Daily;
{$M 8192,,0,,0}

uses Dos;

var Year, Month, Day, DayNo : word;
    FlagFile : file of byte;
    FTime : longint;
    FileTime : DateTime;

function FileExist (NumeFisier : String) : Boolean;
    var Fis : file of byte;
    begin {FileExist}
        {$I-}
        assign (Fis, NumeFisier);
        Reset (Fis);
        {$I+}
        if IOResult = 0 then FileExist := True
        else FileExist := False;
    end; {FileExist}

procedure Execute (Nume : string);
    begin {Execute}
        if FileExist (Nume) then
        begin {executie fisier comenzi}
            SwapVectors;
            exec (GetEnv ('COMSPEC'), ' /C' + Nume);
            if DosError < > 0 then begin
                writeln ('Eroare DOS: ',D osError, ' la executia fisierului ', Nume);
            end;
            SwapVectors;
        end
        else begin
            writeln ('Fisierul de comenzi ', Nume, ' nu exista.');
        end;
    end; {Execute}

begin {Daily}
    GetDate (Year, Month, Day, DayNo);
    assign (FlagFile, 'C:\DATE.DAT');
    if FileExist ('C:\DATE.DAT') then {obtine data fisierului}
        begin
            reset (FlagFile);
            GetFTime (FlagFile, FTime);
            UnPackTime (FTime, FileTime);
        end
    else {initializeaza data}
        begin
            FileTime.Year := 1980;
            FileTime.Month := 1;
            FileTime.Day := 1;
        end;
    if (FileTime.Month < > Month) or (FileTime.Year < > Year)
        {s-a modificat luna}
        then Execute ('C:\MONTHLY.BAT');
    if DayNo = 3 {este miercuri}
        then Execute ('C:\WEEKLY.BAT');
    if (FileTime.Day < > Day) or (FileTime.Month < > Month) or (FileTime.Year < > Year)
        {s-a modificat ziua}
        then Execute ('C:\EVERYDAY.BAT');
    rewrite (FlagFile); {rescriere fisier, cu data curenta}
end. {Daily}
```

DAILY.PAS

```
@echo off
rem Cautare semnaturi virusi
c:\antivir\scan c:
rem Test harddisk
c:\norton\dt c: /d
```

EVERYDAY.BAT

```
@echo off
rem Cautare virusi cu mai multe programe
c:\antivir\vr c: /q/w/t/s
c:\antivir\antivir
c:\antivir\doctor2
rem ... s.a.m.d.
```

WEEKLY.BAT

```
@echo off
rem Reducerea fragmentarii harddisk-ului
c:\norton\sd c:
rem Stergerea zonelor alocare fisierelor sterse
c:\norton\wipedisk c: /e
```

MONTHLY.BAT

Lista modemurilor avizate de Ministerul Comunicaţiilor

Nr. crt.	Tipul modemului	Firma furnizoare	Ţara	Debit binar (bit/s)
1.	GH 2052	ITT	USA	600/1200
2.	MD/6/12	ARE	Italia	600/1200
3.	IBM-3976	IBM	SUA	200
4.	IPA	IPA	România	600/1200
5.	MD-6-12	RACAL MILGO	Anglia	600/1200
6.	TAM-600	TERTA	Ungaria	600/1200
7.	FEA 200	FEA - Bucureşti	România	200
8.	S 22581-F20	Siemens	Germania	600/1200
9.	TRADAN 1300	ITT	Anglia	2400
10.	GH 1151	ITT	Suedia	200
11.	GH 2054-M16	ITT	Suedia	1200/2400
12.	POTOK-AK	-	URSS	600/1200
13.	DS 9322-31A	NOKIA	Finlanda	1200/2400
14.	GH 2054C-16; C-20; F-26; K-16	ITT	Suedia	1200/2400
15.	GH 2058	ITT	Suedia	4800
16.	THC-598	THOMSON-CSF	Franţa	300
17.	2200/24	RACAL MILGO	Anglia	2400
18.	2200/24	RACAL MILGO	Anglia	1200/2400
19.	TELEROM M18	IPA	România	600/1200
20.	MD 200-2	FEA	România	200
21.	TELEROM 3-M1	FEA	România	600/1200
22.	26 LSI	RACAL MILGO	SUA-Austria	2400/1200
23.	MD-101	ROBOTRON	Germania	200 Bd
24.	Cuplor acustic DC-2	Intrep. Echip. Periferice	România	200 Bd
25.	DARO-1641 SSW-GDB CUPLOR	RFT	Germania	9000 Bd
26.	MPS 48	RACAL MILGO	SUA-Anglia	4800 Bd
27.	COM-LINK II	RACAL MILGO	SUA	2400/4800
28.	TELEROM 3MO	FEA	România	300
29.	M 122	IEA	România	600/1200
30.	TELEROM 3M0	IEIA Cluj-Napoca	România	200/300
31.	TELEROM 3M1	IEIA Cluj-Napoca	România	600/1200
32.	MODEM 2084 ITT	ITT	Suedia	2400/1200
33.	MODEM 2082 ITT	ITT	Suedia	600/1200
34.	MODEM PENRIL 4800 D	PENRIL-CORP SUA	SUA	2400/4800
35.	MODEM 3M5	IEIA Cluj-Napoca	România	600/19200
36.	TAM 601	TERTA	Ungaria	600/1200
37.	TELEROM 3M2.1	IEIA Cluj-Napoca	România	2400/1200
38.	2400 P.A. FULL-DUPLEX MODEM	RACAL VADIC	SUA	2400/1200/300
39.	TELEROM M11	IPA Microproducţie	România	600/1200 cu canal retur 75
40.	SE 1801/2/3	-	Anglia	1200
41.	MODEM SISTEM XB	ICPTTC-ICRET	România	600
42.	TELEROM M3	IEIA Cluj-Napoca	România	4800
43.	HAYES 1200B	-	-	1200
44.	ECM 4896M	-	-	4800/9600

Poşta redacţiei

Mă cheamă Mugur, sînt elev în clasa a X-a la lic. "Spiru Haret", secţia informatică. Lucrez de 4 ani pe o Cobra, mai mult în BASIC şi în ultimul an şi în CP/M. În curînd voi avea un IBM AT/286. Ei bine, deşi am o mare dragoste pur. informatică şi "înghit" zilnic o mulţime de informaţii, nu cred că revista "if" mă va ajuta prea mult în învăţarea lucrului cu P.C.-ul. De aceea, VĂ ROG, încercaţi să renunţaţi în cîteva pagini la stilul Dvs. "academic" şi rezervaţi-ne şi nouă, începătorilor, un spaţiu cît de mic.

***

Nu mă va lua cineva la întrebări fiindcă folosesc un produs Borland fără autorizaţie ? Poate publicaţi ceva în legătură cu situaţia de la noi (prezentă şi viitoare) privitor la drepturile de autor în informatică.

ing. Sava Stan, Buzău

Aş vrea să învăţ în primul rînd Pascal şi C şi de ce nu şi limbajul de asamblare MACRO, dar pentru aceasta îmi trebuie ceva documentaţie, pe care nu am de unde s-o iau. Dacă cumva puteţi să mă ajutaţi în acest sens, sînt dispus să cumpăr absolut orice în acest domeniu.

Bogdan Duşmanu, str. Conductei nr. 45, bl. O13, apt. 39, sect. 6, 77586 Bucureşti

Referitor la cîteva articole şi scrisori de la "Poşta redacţiei" vă aduc cîteva completări.

1. În numerele 3 şi 4/91 există două articole referitoare la formatele grafice şi programele de conversie a fişierelor grafice. Pentru cei interesaţi aş mai adăuga cîteva exemple (fără nici o şansă însă de a epuiza întreaga listă existentă).

SCD - prescurtarea pentru SCODL, în programe care se ocupă de tratarea diapozitivelor (Corel Draw, Diapo Agfo Matrix).

AMF - prescurtare de la Aldus Metafile folosit în Page Maker, Pixie 2.

LGO - prescurtare de la LOGO este formatul utilizat de programul de creere/utilizare formulare Jetform.

Full Paint care alături de Mac Paint este utilizat de programul de tratament de texte Quark XPress.

IMA al firmei Zenographics pentru Mirage.

WMF - pentru Docile Album

ASC - pentru ZZ - 2D

DEF - pentru Dyna CADD

CUT - pentru Dr. Halo

GERBER - pentru fotoimprimantă

Două completări: la programul Auto CAD, pe lîngă formatele DXF, SLD, ADI mai există şi RND; iar formatul PIC poate fi întîlnit şi în programe mai noi ca: Video Show, Draw Plus, Pixie, Ami Pro.

În domeniul shareware există numeroase programe care permit vizualizarea, conversia fişierelor grafice cît şi a imaginilor, imprimarea. Exemple:

VGAZEGA V 4.0 - converteşte în imagini EGA (GIF, 16 culori) toate imaginile GIF sau VGA/MCGA în format BLD.

VGAZCGA V 2.0 - converteşte imagini din format GIF în imagini CGA 4 sau 16 culori în format GIF sau CGX.

SQZGIF - converteşte imagini din format GIF (rezoluţie 2048x2048x256) în format VGA/MCGA şi le salvează în standard Blood sau PLT.

MACZGIF - converteşte imagini Mac în imagini GIF sau format Blood.

GIFPUB - converteşte imagini color GIF în imagini alb/negru în format PCX pentru a le încorpora în programe de publicistică stil Ventura.

EGA2VGA V 3.0 - converteşte imagini EGA în VGA înaltă rezoluţie sau în imagini MCGA (300x200x256 cm) format GIF sau CGX.

GIFTOPCX - converteşte fişiere GIF în PCX şi invers

MVGAVU V 1.0b - Afişează pe ecran imagini GIF, BLD şi PLT avînd rezoluţia maximă de 2048 x 2048 x 256 adaptîndule automat la rezoluţia maximă a ecranului (800x600)

GRAPHIC WORKSHOP V 4.3 - permite vizualizarea, conversia, manipularea, imprimarea imaginilor în format Mac Paint, PCX, GEM, IMG, TIFF, GIF, EPS şi MS-Paint. Ieşirea se poate face pe imprimante matriceale compatibile FX-80, pe HP Laser Jet sau Post Script. Memoria EMS sau extinsă, dacă este prezentă, este recunoscută. Programul recunoaşte interfeţele grafice de la standardul CGA pînă la Super VGA.

...

3. Deşi nu este un mouse propriu-zis (5/91), Ballpoint (trackball) este ultimul produs al firmei Microsoft în domeniu. Este rezervat calculatoarelor portabile şi se poate ataşa lateral (stînga sau dreapta, la dorinţa utilizatorului) cu ajutorul a două cleme metalice. În partea exterioară are patru butoane care pot fi apăsate atît cu degetele cît şi cu podul palmei. Este livrat cu un nou program Mouse.com v. 8.0, compatibil cu toate celelalte mouse-uri şi programe Microsoft. El permite modificarea vitezei cursorului, iar sub Windows se pot modifica şi funcţiile tastelor şi axele de deplasare.

Tot pentru amatori de sisteme cu bilă, firma Future Soft a lansat pentru Macintosh, PC şi portabile un nou produs Thumbelina (194$) care este o minusculă trackball ce se ţine în mînă.

O inovaţie în domeniu deţine firma americană Appoint prin noul produs Mousepen. El a constituit obiectul a numeroase brevete, în special pentru sistemul miniatural de fricţiune în punctul de contact. Sub formă de stilou el conţine în vîrf o bilă mică, dură şi rezistentă. Pe partea din faţă, două butoane sînt aşezate la nivelul indexului.

În greutate de 90g, cu un consum electric minim, cu un spaţiu de lucru necesar de doar cinci cm^2 el poate lucra pe orice material sau suprafaţă. Spre deosebire de mouse-urile clasice, Mousepen nu are nevoie de nici o întreţinere deosebită. Se poate conecta direct la portul ADB a unui Macintosh, la interfaţa serie a unui PC sau a unui portabil (RS-232C/1200 bds). Pentru Macintosh nu este necesar nici un program. Pentru PC, Mousepen dispune de un pilot de reglare a vitezei. Este compatibil cu mouse-urile Microsoft şi cu mediul Windows. În versiunea Pro, Mousepen prezintă în plus un al treilea buton care permite ajustarea în timpul execuţiei unei aplicaţii - tratament de texte, desen de mare precizie - a fineţei rezoluţiei vîrfului. Preţul pentru PC este 209$ iar pentru Macintosh 193$.

4. Versiunea "beta" DOS 5.0 (5/91) a fost testată de mai mult de 7000 de utilizatori nepunînd nici o problemă de "etanşitate". Secretul, altfel bine păstrat, a fost de multe ori deconspirat nu numai de articolele şi notele din presa americană ci şi de "experţii", previzioniştii şi responsabilii micro din numeroase mari întreprinderi. De la versiunile "beta" 333 pînă la 409c niciodată nu au circulat un aşa număr mare de "transmisii nedelicate".

DOS 5.0 ca şi DR-DOS a devenit mai "prietenos" permiţînd afişarea sintaxei comenzilor. În lucrul cu harddisk-ul programul cache Smartdrive a fost înlocuit cu un program puternic Fast! al firmei Future Computer. Suprafaţa utilizator Dosshell-semigrafică, poate genera ecrane VGA avînd aceaşi capacitate de gestionare a fişierelor ca şi Windows în ceea ce priveşte deplasarea, copierea sau ştergerea documentelor fiind un concurent puternic a lui Norton Comander. Este permisă, pentru prima dată, formatarea dischetelor de 3 1/2" la 2,88 Mo. (Sînt aşteptate , pe piaţă, noile lectoare de discuri de 3 1/2" - 2,88 Mo HD).

Se poate spune despre DOS 5.0 că este: o versiune 3.31 încorporînd un DOS Extender, ca şi Qemm al firmei Quarterdeck; permite afişarea sintaxei comenzilor, ca şi DR-DOS al firmei Digital Research, are un Dosshell, ca şi Progman sub Windows 3.0 sau ca şi Norton Commander; are o instalare automată, ca toate celelalte programe; are în sfîrşit utilitare de manipulare a fişierelor, ca şi PC Tools al firmei Central Point Software - de fapt, aceleaşi: Microsoft a cumpărat licenţa de la CPS.

5. Pentru iubitorii familiei de calculatoare Macintosh şi pentru amatorii de date exacte am corectat şi completat datele articolului din ultimul număr referitoare la acestea.

Classic are frecvenţa de ceas 7,8336 MHz, iar SE/30 şi Portable au frecvenţa 15,6672 MHz. SE/30 are 2560 de memorie cache; //ci şi //fx au 256 o de memorie cache pentru date şi 256 o de memorie cache pentru instrucţiuni. Ultimele două datorită RAM-ului cu timpul de acces de 80 ns permit paginarea rapidă ("burst mode"). Nu numai //fx cît şi celelalte două calculatoare din familia constructivă modulară //si şi //ci cît şi LC au 512 ko ROM care conţin printre altele şi programul Quick Draw pe 324 ceea ce le permite afişarea simultană a 16,8 milioane de culori.

//si, //ci şi //fx pot lucra opţional cu tastatura extinsă de 106 taste. Opţional primele două pot utiliza un harddisk de 80 Mo iar //fx unul de 160 Mo.

//ci are un conector de RAM-cache 120 de contacte.

//fx are 32 ko de memorie cache integrată - SRAM (RAM static) cu timpul de acces de 25 ns ce utilizează Fast Memory Controller.

Tot //fx are un conector cu 120 de contacte care permite accesul direct la bus-ul microprocesorului putîndu-se astfel accelera viteza calculatorului prin interfeţe de mare viteză.

Classic, LC şi Portable dispun de un circuit sonor ASC (Apple Sound Chip) ce poate genera patru căi pe 8 biţi la 22 kHz; la celelalte ASC conţine un demodulator stereo pe 8 bits la 44,1 KHz şi un sintetizator quadrofonic permiţînd utilizarea căştilor sau a boxelor stereo.

Classic nu are intrare audio ci doar ieşire mono.

SE/30 - port ieşire stereo

LC - port intrare mono, port ieşire mono

//si - port intrare mono, port ieşire stereo

//ci şi //fx - doar port ieşire stereo

Portable este identic cu Classic. El mai poate fi livrat cu taste numerice în loc de "trackball".

Referitor la memoria acestor calculatoare cît şi la lectorul SuperDrive se poate consulta Poşta redacţiei - if 4/91.

6. Punctul acesta nu are nimic de a face cu vreun articol din revistă. M-am hotărît să-l scriu doar după emisiunea "Universul cunoaşterii" de ieri, miercuri 17.07.91 ora 22.15.

În debutul emisiunii; un reporter care a mai lucrat "cîndva" în informatică ne prezenta un calculator "specializat" în sinteza vocală. Eu cînd aud vorbindu-se de domeniul respectiv imediat mă gîndesc la un laborator de cercetare şi la nişte preţuri pe măsură. Nici vorbă de aşa ceva. Cred că la fel ar fi vorbit şi un profan în domeniul calculatoarelor care văzînd un PC cu un program de grafică l-ar fi declarat "specializat" în desenare. Este regretabil ca într-o emisiune care se vrea de informare ştiinţifică, un reporter "să vîndă" telespectatorilor tot ceea ce i se oferă, fără nici un discernămînt.

Totuşi să vedem despre ce este vorba.

SOUND BLASTER este o interfaţă sonoră care reuneşte 11 căi FM de sintetizare (compatibile ADLIB) şi un canal de digitizare vocală; are integrat un amplificator de putere, y ohms, y watis pe canal cu posibilitatea reglării volumului printr-un potentiometru exterior; are de asemenea integrat un amplificator pentru digitizarea analogică pe 8 bits (5 - 13 KHz/ mod direct sau transfer DMA) cu o intrare micro (600 dms); mai este dotată cu un port joystick care poate fi folosit şi pentru o interfaţă MIDI şi cu o ieşire cale numerizată (stereo) pentru boxe, casă sau amplificator stereo.

SOUND BLASTER se poate implanta în oricare din slot-urile de extensie ale unui PC, XT, AT, PS/2. Configuraţia necesară funcţionării: minim 512 ko RAM, DOS 2.0 sau o versiune ulterioară, orice interfaţă video MG, CGA, EGA sau VGA.

Interfaţa este livrată cu o dischetă demonstrativă cu efecte sonore, un cablu pentru amplificator şi cinci programe. (Două au putut fi urmărite la TV). Preţul 499 DM.

TALKING PARROT - reprezintă un papagal care nu suportă tastatura răspunzînd prin strigăte; acceptă doar microfonul prin care utilizatorul poate comuunica cu el, imitîndu-i vocea; imaginea şi sunetul pot fi modificate. Pentru o mai bună calitate este de preferat un AT.

FM ORGAN - prin intermediul acestui program se poate cînta sau învăţa să se cînte pe tastatura PC-ului. Este un sintetizator monofonic care oferă: la alegere 20 de instrumente, 80 de acompaniamente pre-înregistrate, posibilitatea modificării tempo-ului, ritmurilor. Se pot utiliza şi instrumente MIDI, care se conectează prin intermediul unei interfeţe MIDI (optional) la portul joystik al interfeţei.

VOXKIT - permite înregistrarea, stocarea şi ascultarea oricărui sunet, cît şi compresia şi decompresia fişierelor.

SB TALKER - este un sintetizator text-voce care face posibilă introducerea unui text care apoi va fi citit de calculator cu pronunţie americană.

DR SBAITSO - este un program de inteligenţă artificială cu care se poate conversa în engleză asupra diferitelor subiecte.

Opţional există o interfaţă MIDI şi softul aferent care permite utilizarea instrumentelor MIDI (orgă, sintetizator...) în direct cu calculatorul (înregistrări, reascultări...).

Tot opţional există boxe "style walkman" care se conectează direct în prizxa jack de ieşire. Sînt livrate de asemenea două chip-uri C/MS care permit exploatarea jocurilor şi programelor c/MS şi GAME BLASTER. Ele se conectează în cele două socluri libere de pe interfaţă. Preţul lor este de 39,95 DM.

Trebuie menţionat că există numeroase jocuri care utilizează posibilităţile sonore ale interfeţei SOUND BLASTER. Ex: Battle of Britain, Indiana Jones & the Lost crusade, The secret of markey island, Prince of Persia, Lorols of the rings, Bard's tale III, Ghostbuster II, Time of Lore, Budakan, etc.

7. Sper să le fac o bucurie iubitorilor de Macintosh prin următoarea adresă:

Apple Computer France
12, avenue d'Oceanie
Z.A. de Courtaboeuf
91956 Les Wis Cedex.

Voicu Manole, str. Măcieşilor 1, 1900 Timişoara, tel. 961/35037

Mica publicitate

Firmă particulară cu experienţă în domeniu, execută **programe de evidenţă financiară, contabilitate**, pe calculatoare compatibile IBM - PC. Telefon 90/17.80.95

Oferim dezvoltări software : **simulări procese industriale, autotestări cunoştinţe, interfeţe interactive.** G.Anescu, ICN, C.P.78, 0300 Piteşti

Vînd calculator personal CIP, 80k, compatibil **ZX - Spectrum**, în stare de funcţionare. Preţ: 15.000, tel. 90.16.78.13 (seara).

Caut **sursă program sinteză vocală**, orice compatibilitate (preferabil IBM-PC). Tel. 916/88660.

Cumpăr originale sau xerox-uri din domeniile: **grafica prin raytracing; formate grafice pt. fişiere**. Doresc descriere carte sau cuprins. Ciocîrlan Florin, Brazda lui Novac, F9, 1, 19, 1100 Craiova, Tel. 941/54988.

Şi Dumneavoastră puteţi contribui la îmbunătăţirea conţinutului revistei ! Ajutaţi-ne, completînd chestionarul alăturat, şi revista se va apropia mai mult de ceea ce aşteptaţi de la o revistă destinată utilizatorilor de PC-uri ! Vă mulţumim !

Chestionar

1. Cel mai mult mi-au plăcut următoarele articole:
   1. _____
   2. _____
   3. _____
2. Cel mai puţin mi-au plăcut următoarele articole:
   1. _____
   2. _____
   3. _____
3. În numerele următoare aş dori să fie incluse articole care să trateze:
   1. _____
   2. _____
   3. _____
4. Utilizez un calculator:
   |_| acasă |_| la birou
5. Tipul calculatorului utilizat:
   _____
6. Lucrez mai frecvent cu următoarele pachete software:
   1. _____
   2. _____
   3. _____
7. P.C-urile mă interesează ca:
   |_| profesionist |_| hobby
   |_| student |_| elev
   |_| utilizator |_| posibil utilizator
8. Citesc revista "if":
   |_| regulat |_| ocazional
9. Ce-ar mai fi de spus ? _____

Data: _____ Semnătura: _____ **If 7 / 91**

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 172, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if,** pe o perioadă de |_____| apariţii, pe adresa:

Nume&prenume .....

Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....

Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 69 lei. Preţul unui abonament se va calcula cu formula: 69 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

if 7 / 91 **Data: ..... Semnătura .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

Important !

Doriţi să aveţi garanţia că nu veţi pierde nici un număr din "**if**" ?
Încheiaţi un abonament folosindu-vă de talonul alăturat,
sau adresaţi-vă difuzorilor noştri autorizaţi:

Arad
Boros Ladislau
tel. 73412, 31778

Bacău
Pătru Liviu
str. 9 Mai, bl. 36, sc. B, ap. 15

Baia Mare
Omni Computer
Cudalbu Octavian
tel. 36433

Brăila
Teodorescu Constantin
tel. 47203

Bucureşti
Dracea Tudor
tel. 210340

Buzău
S.S.I. Infobuz - Buzău S.A.
ing. Ursu Ioan
tel. 31844

Cluj Napoca
AbMod
- Magazin Hardware-Software b-dul 22 Decembrie nr. 135
- Sediul central tel. 111090

Costin Suciu Libris
tel. 145526

Craiova
Stătescu Paul
tel. 48117

Deva
Toma Tudor
tel. 10144, 10234

Drobeta Turnu-Severin
Bazavan Constantin
tel. 23666 - 143

Oneşti
Brumă Cornel
tel. 24222-236

Oradea
AbMod
tel. 60278, 44476

Rădăuţi
Burciu Florin
tel. 63483

Sighişoara
Beer Mihai
tel. 73638

Timişoara
Tomoroga Mircea
tel. 77422

Căutăm difuzori/distribuitori autorizaţi
pentru difuzarea revistei "**if**" !.
Condiţii avantajoase !
Rugăm persoanele interesate să ia legătura cu redacţia.

μmicro ATCI S.R.L.
Tîrgu Mureş
str. Gheorghe Doja nr. 36
tel. 954/31660

MenuPdx 1.0

Bibliotecă de meniuri pentru Paradox 3.5

- Cuprinde:
  - 6 modele de meniuri Pull Down
  - 8 modele de meniuri Pop Up

scrise în PAL şi apelabile direct din aplicaţia D-voastră.
- Realizează înlănţuirea automată a meniurilor şi lansarea procedurilor din aplicaţia D-voastră; tot ce vă rămîne de făcut este crearea unei/unor tabele de meniuri, (eventual help-uri) şi să scrieţi procedurile care vă rezolvă problema.
- Demo disponibil.
- Preţ informativ: 4895 lei.

TIMPUL DUMNEAVOASTRĂ ESTE PREŢIOS !
CONTACTAŢI-NE ŞI VEŢI SCĂPA DE MUNCA DE RUTINĂ !

Societatea de Informatică **DELTA-X S.R.L.**
Oradea, P-ţa Independenţei Nr.47 A3/23
Telefon: 991-34257, 51804

Vă oferă la preţuri avantajoase următoarele produse program generalizabile pe calculatoare compatibile IBM PC XT/AT:

1. Bal-Con - evidenţa contabilă generală

2. Mi-Fi - evidenţa şi calculul amortizării mijloacelor fixe

3. Ge-Ma - gestiunea şi evidenţa contabilă a valorilor materiale (materiale, obiecte de inventar, produse finite).

Produsele enumerate, sînt puternic parametrizate avînd un înalt grad de generalizare şi au strînse legături între ele, utilizînd colecţii comune de date, transferînd informaţii între ele. Evidenţa contabilă Bal-Con, reprezintă "vîrful piramidei" care pune la dispoziţia celorlalte produse planul de conturi contabile, respectiv primeşte date (note contabile generate automat) din partea acestora. Sistemul este deschis, nefiind obligatorie utilizarea produselor program enumerate împreună, respectiv este facilitată conectarea altor aplicaţii la evidenţa contabilă (salarii, desfacere, etc.). Utilizarea şi operarea este facilă prin meniurile, mesajele explicative concrete apelabile în toate fazele de prelucrare, şi autodocumentare.

Prin programele de instalare livrate, utilizatorul poate configura produsele la caracteristicile echipamentului utilizat.

Versiunile finalizate ale celor trei produse menţionate reflectă legislaţia economică şi reglementările metodologice în vigoare şi vor fi aliniate şi în viitor la eventualele schimbări legislative sau metodologice care vor apare.

4. dAccess - accesare multiuser a bazelor de date compatibile dBASE.

Permite culegerea, prelucrarea şi vizualizarea datelor în regim multiuser, de la 1 la 8 terminale, conectate pe căile seriale COM1 - COM8 şi de la tastatura/monitorul calculatorului.

Oferă în regim multiuser, prin taskuri independente, majoritatea prelucrărilor permise de sistemele tip dBASE, definirea acestor taskuri făcîndu-se într-un mod simplu, familiar utilizatorilor.

Avînd în vedere, că majoritatea utilizatorilor de tehnică de calcul din economia noastră, dispun de terminale simple în bună stare de funcţionare (VDT-52S, etc.), prin utilizarea acestui produs achiziţionarea unor calculatoare IBM-PC compatibile în configuraţia obişnuită (2 căi seriale) nu exclude utilizarea eficentă a acestor terminale devenite disponibile.

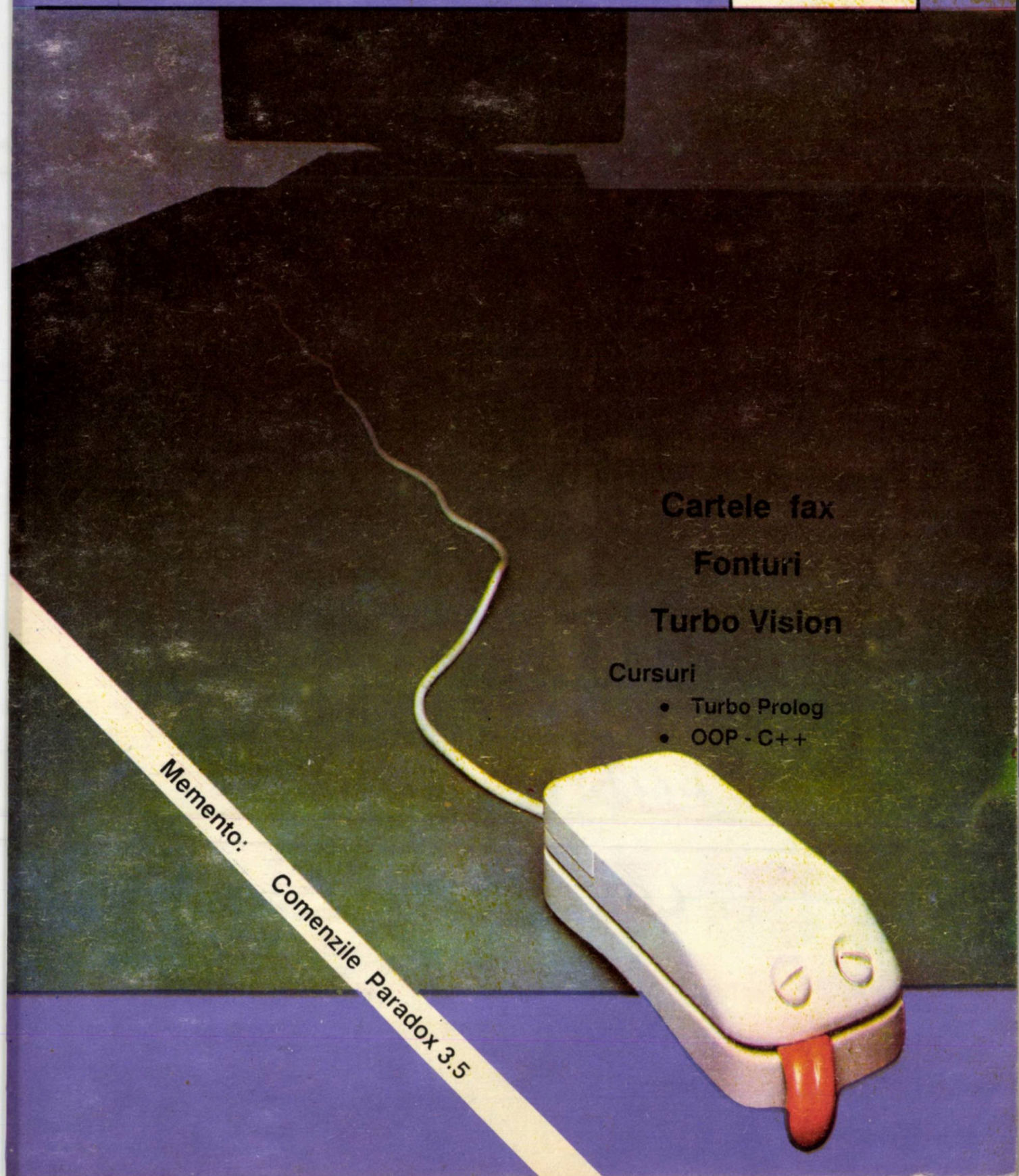
Tiparul executat la Tipomur S.A. **C.D. Nr. 78 1991 Tiraj: 10.000 ex.**
Tirajele anterioare: 1/90 - 5.130, 2/90 - 5.000, 3/90 - 5.000,
1/91 - 7.365, 2/91 - 7.500, 3/91 - 7.620, 4/91 - 8500, 5/91 - 10.000, 6/91 - 10.000

if

CALCULATOARE PERSONALE

Revistă lunară editată de Micro ATCI S.R.L. Tîrgu Mureş

8 / 91
(11)
125 Lei
ISSN 1220-1529



Cartele fax

Fonturi

Turbo Vision

Cursuri

- Turbo Prolog
- OOP - C++

Memento: Comenzile Paradox 3.5

- **Doriți** mijloace standard de conectare la baze de date din Europa?
- **Doriți** interconectarea calculatoarelor dumneavoastră?
- **Vreți** transferuri de date între diverse tipuri de calculatoare?
- **Aveți** echipamente cărora doriți să le atribuiți valențe noi?
- **Doriți** o linie tehnologică completă pentru editare cu caractere în alfabet românesc?
- Vă este necesar un sistem de introducere și verificare date multiacces pe PC, adaptabil la aplicațiile dumneavoastră?

Experiența și realizările laboratorului reprezintă cartea noastră de vizită:

- emulatoare transfer fișiere în comunicație sincronă compatibile IBM 2780/3780 pe Felix, mini, micro 8 biți și Robotron 1834 (sau orice PC care are comunicație sincronă);
- concentrator de terminale IBM 3271(mini,micro);
- linia ARIEL (ARIEL, EMMA, emma);
- acces și transfer fișiere multiacces mini-micro în comunicație sincronă (EMMI, emmi);
- editor de fonte VENTURA, fonte românești pentru VENTURA și WORDSTAR, procesoare de texte;
- software pentru conectarea calculatoarelor la o rețea X25;
- PAD-X25 - asamblor/dezasamblor de pachete într-o rețea de calculatoare X25;
- IDA - sistem data-entry multiuser pentru PC cu instrumente de dezvoltare aplicații specifice Data-entry;
- instalări de rețele locale de tip Novell.

NU EZITAȚI!

CONTACTAȚI:

ICI - Laborator **SYSNET**
B-dul G-ral Averescu, Nr.8-10
tel. 65.60.60/123,118, cam. 105,
112

if

revistă de informatică
editată de firma **Micro ATCI**

Director: ing. Dumitru Dunca

Redacţia:
ing. Iosif Fettich,
ing. Ingrid Maier,
ing. Romulus Maier,

Colaboratori externi:
ing. fiz. George Anescu,
stud. Daniel Buleu,
ing. Alin Flaidăr,
ing. Stelian Florea,
ing. Răzvan Grigorescu,
ing. Cristian Malide,
ing. Voicu Manole,
ing. Francisc Publik,
ing. Stan Sava
stud. Tiberiu Socaciu,
cercetător Pandele Stănescu

Tiparul: tipografia Tîrgu Mureş

Revista apare lunar.

Preţ: 69 lei

Manuscrise originale sau listinguri de programe sînt primite cu plăcere de redacţie, cu condiţia să nu fi fost publicate şi în altă parte. Prin expedierea unui manuscris pe adresa redacţiei, autorul consimte implicit la publicarea materialului său în cadrul revistei. Onorariul se negociază cu redacţia. Materialele nepublicate nu se înapoiază şi nu se reţin.

Revista noastră vă oferă spaţiu pentru reclamă şi publicitate. Pentru amănunte vă rugăm să luaţi legătura cu redacţia.

Cei care doresc să anexeze revistei pliante publicitare tipărite în regie proprie, sînt rugaţi şi ei să se adreseze redacţiei .

Adresa şi telefonul redacţiei:
Micro ATCI, redacţia "if",
RO-4300 Tîrgu Mureş,
C.P. 172, O.P. 1,
tel./fax 954/31660 (direct),
33612, 24158, 20057, 33511,
int. 134 sau 189
41417 sau 46885 (după ora 19)
telex 65354 (Intertours)

Happy Aniversary !

Firile sentimentale au sărbătorit în august împlinirea a 10 ani de la naşterea calculatoarelor personale. În august 1981 IBM prezenta primul calculator personal. Aceşti 10 ani au fost caracterizaţi de o dezvoltare în ritm susţinut a hardware-ului, ritm care cel mai adesea nu a putut fi urmărit şi de dezvoltarea software-ului.

În următorii 10 ani situaţia se va schimba. În centrul atenţiei se află trei domenii: comunicaţiile, grafica şi software-ul orientat obiect.

Comunicaţii: Se estimează că în următorii 3(!) ani fiecare PC va fi cuplat într-o reţea. Problema legăturilor fizice este rezolvată (în alte ţări de soare pline), mai rămîne de rezolvat problema soft-ului. Se mai estimează că în curînd poşta electronică (E-Mail) va juca un rol mai important decît transmisia prin fax. Pînă atunci puteţi folosi însă calculatorul Dvs. şi ca aparat fax. Cum? Veţi afla din articolul prezentat în acest număr.

Grafică: Şi în acest domeniu se vor realiza progrese semnificative. Cel mai important domeniu de utilizare a graficii pe calculator va fi în cadrul aplicaţiilor multimedia. Încă la sfîrşitul acestui an va fi prezentat Windows 3.1 care va avea o extensie multimedia. Un "Multimedia-Postscipt" va rezolva problema diferitelor formate ale datelor. Despre multimedia am mai vorbit şi vom mai reveni. În acest număr vă prezentăm, în articolul despre fonturi, problemele tehnice care pot să apară la tehnoredactarea pe calculator. Facilităţile grafice sînt esenţiale şi în acest caz.

Software orientat obiect: Aceasta nu înseamnă altceva decît că utilizatorii vor fi în măsură să-şi ajusteze singuri aplicaţiile. Aplicaţiile vor deveni scalabile şi vor fi livrate sub forma unor biblioteci de obiecte urmînd ca utilizatorii să-şi definească un mediu de lucru "personal". Sperăm să deschidem o fereastră spre astfel de aplicaţii prezentîndu-vă produsul Turbo Vision al firmei Borland şi antamînd un curs de C++.

În acest număr facem şi cîteva dezvăluiri senzaţionale - citiţi articolul "Cine eşti dumneata Bill Gates, şi ce-ai făcut în ultimii 5 ani?", intitulat de fapt: "Factorul 20%".

Lectură plăcută!

Romulus Maier

Cuprins

Ştiri

- Noutăţi pe piaţa produselor software pag.4
- Instant Logic pag.4
- NTT cu display 3D! pag.4
- Un nou harddisk Seagate pag.4
- Powersave 500 . . . . . pag.5
- A doua generaţie de programe antivirus pag.5
- Dublaţi capacitatea discului Winchester ! pag.6
- Expoziţie pag.6

Profil

Factorul 20% pag.7

O legendă vie: Bill Gates

Strategia Microsoft pag.8

Windows here, Windows there, Windows everywhere

Puncte de vedere

Protecţia programelor pag.10

Comunicaţii

PC sau telefax? - Cartele fax . . . . . pag. 13

Ca alternativă la telecopiator ca echipament separat, PC-ul - dotat cu hard şi soft corespunzător - cîştigă din ce în ce mai mult teren ca emiţător şi receptor de telecopii. "if" vă prezintă avantajele şi dezavantajele cartelelor fax.

Fonturi

Scrierile şi calculatorul pag.20

Generatoarele de fonturi pag.23

Emulatoarele Postscript pag.25

6 din cele mai importante soluţii hard . pag.28

Software

Turbo Vision pag.30

În primăvara acestui an profetul limbajelor de programare orientate obiect, Borland, a împărtăşit credincioşilor săi ultima sa viziune. Turbo Vision nu este doar o colecţie de obiecte ce trasează cadrul general al unei aplicaţii, ci este şi suportul unui concept nou în programarea PC-urilor. Deşi timpul de iniţiere a crescut, a scăzut dramatic durata finisării unei aplicaţii iar cizelarea ei poate fi realizată impecabil.

Cursuri

Turbo Prolog (partea a 2-a) pag.36

După introducerea în principiile de bază ale programării logice din episodul anterior, de data aceasta vom examina tehnica de programare în Turbo-Prolog, prin intermediul unui exemplu.

OOP: C++ - (partea întîi) pag.40

Nici o filozofie de programare nu a stîrnit atîta interes în ultimul timp ca programarea orientată obiect. După cursul introductiv pentru limbajul Pascal, vă prezentăm acum continuarea pentru cei interesaţi de C++.

Reţele Novell

Novell Lite v1.0 . . . . . pag.46

Practică

- Parolă variabilă pag.50
- 1024 . . . . . pag.51
- Cîte ceva despre tastatura AT pag.51
- O glumă pag.52
- Recondiţionarea riboanelor pentru imprimantă pag.53

Rubrici

- Caseta redacţiei . . . . . pag. 1
- Editorial . . . . . pag. 1
- Poşta redacţiei pag.54
- Mica publicitate pag.54

Noutăţi pe piaţa produselor software

LOGIC, în calitate de distribuitor în România, a fost anunţat că la începutul lunii iulie 1991, **Borland** a încheiat un acord pentru **achiziţionarea Corporaţiei Ashton Tate.**

Borland va continua să actualizeze şi să dezvolte produsele **Paradox Borland** şi **dBase Ashton Tate** pentru sistemul de operare DOS. Mai mult, Borland va continua să ofere dBase IV pentru mai multe sisteme de operare, incluzînd UNIX 386, SUN OS, DEC VAX VMS şi Macintosh.

Borland este în curs de elaborare a unei tehnologii pentru interpretor şi compilator dBase pentru utilizatori DOS, Windows şi pentru alte sisteme de operare.

Noile produse Windows, numite **"Paradox for Windows"** şi **"Object dBASE"** vor fi compatibile cu omologele lor DOS. Produsele vor fi realizate pe baza noii arhitecturi **Borland engine**, permiţînd interschimbarea transparentă a datelor între Paradox, produsele **dBASE, Btrieve, spreadsheet** şi **SQL**. Data de punere în vînzare nu a fost anunţată.

Produsele Windows vor fi bazate pe tehnologia Borland evoluată, orientată pe obiecte. Acestea vor fi capabile să gestioneze obiecte cum ar fi forme, rapoarte sau cereri. Aceste noi versiuni ale produselor dBASE şi Paradox vor fi complet interoperabile cu produsele Borland de vîrf de pe piaţă, incluzînd **Object Vision** şi **Borland C++**.

Borland va continua să susţină şi să dezvolte produsele **Ashton Tate Interbase** pentru sistemele existente, incluzînd UNIX şi VMS.

De asemenea Borland plănuieşte să asigure susţinerea în continuare şi a altor produse Ashton Tate şi va dezvălui detalii despre acele produse după terminarea integrării.

Definitivarea unirii complete între Borland şi Ashton Tate este dependentă, pe lîngă alte lucruri, de acordul acţionarilor fiecărei companii şi de obţinerea tuturor aprobărilor guvernamentale necesare. Tranzacţia se aşteaptă să fie încheiată cel mai tîrziu la sfîrşitul acestui an.

O altă noutate, oferită de firma LOGIC, este oferta de a achiziţiona produsul Borland **Turbo Pascal Professional 5.5 la un preţ de numai 6495 lei.**

Stelian Florea
director - LOGIC Sibiu

Instant Logic

Inginerii şi designerii circuitelor integrate îşi pot face acum sintezele logice şi pe un PC, (în curînd pe orice staţie de lucru), cu ajutorul noului pachet intitulat "Instant Logic" al firmei Autodesign Inc. Pot fi generate logici pentru execuţii logice, tabele de adevăr şi pot fi definite automate finite.

Versiunea 2.0 a produsului "Instant Logic" poate fi rulată pe orice calculator compatibil IBM, de tip PC AT/386, cu cel puţin 640 KB memorie. Un circuit poate să aibă pînă la 300 - 400 de porţi. Versiunile viitoare vor avea incorporate extensii DOS, uşurînd munca de design considerabil.

"Instant Logic" este o colecţie de unelte (tools); incluzînd un sintetizator, cu ajutorul căruia se pot utiliza listele Orcad, execuţii logice, descrieri de automate finite sau tabele de adevăr şi toate acestea pot fi sintetizate pentru implementarea porţilor; un utilizator - analizor al costului şi timpului necesar pentru design; un simulator funcţional; un viewer logic; un program generator de biblioteci pentru construcţia simbolurilor grafice necesare schemelor lui "Instant Logic".

Elementele primitive (porţile şi bistabilii) pot fi descrise cu ajutorul descriptorilor ASCII care specifică ridicările şi coborîrile de întîrziere, perimetrul porţilor şi pinii exteriori ai circuitului anticipat. Oricum sistemul are o bibliotecă cu seturi generice.

Utilizatorii pot rula "Instant Logic" pe orice PC AT standard cu 286 sau 386. Cu un 386 la 20 MHz se compilează cu o rată de 100 porţi pe minut. Soft-ul suportă monitoare color atît EGA, cît şi VGA. "Instant Logic" este disponibil de un an deja la un preţ de 495$.

(S.T.)

NTT cu display 3D!

Nippon Telegraph&Telephon Corp. (NTT) a construit un nou display color pentru imagini tridimensionale fără a utiliza nici o sticlă specială. Ecranul color are o dimensiune de 15 inch şi pe el se proiectează două imagini - una pentru fiecare ochi -, care ajung separat şi direct la fiecare ochi în parte. Rezultatul (din punct de vedere optic) este o singură imagine la care nu mai trebuie să te uiţi cu familiarii ochelari de corecţie. Se pare că singura dificultate este producerea lentilelor (după spusele unui cercetător de la NTT Human Interface Laboratories). NTT combină cele două lentile plate c u un adeziv ultraviolet. Noile display-uri sînt echipate cu un "infra red sensor" care este capabil să "ducă" capul utilizatorului în timp real.

(S.T.)

Un nou harddisk Seagate

În cursul trimestrului trei/91 firma Seagate va lansa pe piaţă STY2400N, un harddisk de 1 1/4" pentru interfaţa SCSI-2.

Capacitatea sa de 2,4 Go va constitui cel mai înalt nivel de stocare în acest format. Preţul de vînzare va fi de aproximativ 3500$.

Trebuie menţionat că Seagate dispune de un harddisk de 426 Mo SCSI-2 în format 3 1/2".

(V.M.)

Powersave 500

Această realizare a firmei Dakota Microsystems împiedică pierderea de date în cazul întreruperii curentului refăcînd starea memoriei la revenirea tensiunii.

Esenţial pentru o sursă neîntreruptibilă este să furnizeze calculatorului curent continuu un anumit timp. Soluţia externă nu este cea mai judicioasă deoarece este necesară trecerea prin două conversii: CA/CC şi CC/CA (curent alternativ/curent continuu şi invers) cu un randament foarte bun, ceea ce se realizează greu din punct de vedere tehnic. Firma Dakota şi-a imaginat plasarea sursei direct în calculator şi a creat cartela Powersave 500 ce poate fi considerată ca o baterie de 12V/1,2A cu reîncărcător încorporat. Un regulator transformă această tensiune în ieşiri +5V/20A, +12V/8A, -5V/5A şi -12V/5A. Sistemul este comutat în cazul în care detectorul de tensiune constată o scădere de la +5V la 4,75V.

Volumul redus ocupat de cartelă nu permite utilizarea unei baterii voluminoase astfel că Powersave 500 poate furniza 200W numai 90 de secunde, timp suficient însă. Pentru întreruperile: pînă la 2 secunde ea înlocuieşte doar sursa de alimentare; între 2 şi 20 de secunde salvează memoria calculatorului. Dacă curentul revine în mai puţin de 30 de secunde atunci va restaura contextul existent anterior, utilizatorul putînd să-şi continue munca. În caz contrar va opri calculatorul rămînînd în aşteptare şi va putea reface vechea stare la revenirea tensiunii.

Punerea în funcţiune a cartelei Powersave 500 este ceva mai complicată decît în cazul unei surse neîntreruptibile externe. În primul rînd este necesar un conector liber situat dacă este posibil cît mai departe de interfaţa controllerului de harddisk din cauza radiaţiilor parazite. Grosimea cartelei face imposibilă utilizarea conectorului alăturat. Montarea cablurilor nu este prea elegantă deoarece fişele P8 şi P9 ale alimentării trebuie mutate la un capăt al cartelei, aceasta aflîndu-se în cele mai neaşteptate locuri în calculator. (Este vorba despre compatibile). Cel mai supărător lucru este însă instalarea soft-ului. Aceasta înseamnă octeţi în minus în memoria de lucru, o linie de transmisie ocupată permanent (COM1, COM2 sau LPT2) şi 32 Ko ocupaţi în high-memory pentru BIOS-ul cartelei Powersave 500.

Refacerea contextului la sfîrşitul unei întreruperi de curent este din fericire parametrizabilă (manual sau automat). Salvarea memoriilor EMS şi video este, de asemenea, prevăzută. Restaurarea corectă în cazul unor programe care lucrează cu intefaţa video într-un mod de emulare face necesară recunoaşterea acestora, complicîndu-se apoi cu apelul unei proceduri DOS specifice acestui mod.

Un lucru pe care trebuie să încercăm să nu-l uităm este prezenţa cartelei în calculator. Înainte de oprirea lui este necesară apăsarea tastelor ALT-D. Cei prea grăbiţi sau cei obişnuiţi cu mijloace de deconectare mai puţin ortodoxe vor avea surpriza, la viitoarea punere în tensiune, de a-şi regăsi ultima aplicaţie la care au lucrat.

Powersave 500 nu este o alimentare suplimentară, de rezervă, ci un dispozitiv care permite utilizatorului recuperarea datelor în cazul întreruperii curentului. De fapt interfaţa face acelaşi lucru ca oricare altă sursă neîntreruptibilă externă oferind în plus şi modul automat ceea ce îi oferă un raport bun calitate/preţ (1050 DM).

(V.M.)

A doua generaţie de programe antivirus

Firma californiană "Xtree" a conceput o a doua generaţie de programe de protecţie împotriva programelor de sabotaj. Programul de protecţie "Virusafe" nu recunoaşte doar viruşii cunoscuţi ai calculatoarelor, ci este capabil să controleze şi potenţialul distructiv al unor programe suspecte (posibili viruşi).

Experţii au din nou păreri diferite. În SUA au fost găsite, pînă în prezent, 500 de variante de programe de sabotare, în schimb profesorul Klaus Brunnstein a numărat de curînd chiar 1000. Zăpăceala intervenită în timpul numărării viruşilor nu trebuie să ne mire. În funcţie de faptul că variantele unor programe de sabotaj cunoscute se încadrează ca viruşi "independenţi" sau se consideră ca "descendenţi" ai viruşilor deja apăruţi, numărul acestora variază între limite foarte largi.

Programul de devirusare al firmei californiene "Xtree" promite o curăţire meticuloasă. Noua versiune a programului lor de devirusare "Virusafe" ar fi în stare să recunoască şi să elimine şi viruşi încă necunoscuţi.

Cea mai mare problemă a luptei contra viruşilor, pînă în prezent, a fost cursa contra timpului. Puhoiul de programe antivirus a şchiopătat mereu în urma inventivităţii criminale a programatorilor de viruşi şi în urma "softurilor degenerate" apărute întîmplător.

Noua versiune a programului "Virusafe" nu recunoaşte doar toţi viruşii cunoscuţi pînă în prezent ci şi noi "variante" ale lor. El nu elimină doar viruşii existenţi, ci deţine şi un sistem de verificare pentru software suspect, care funcţionează în felul următor: atunci cînd "Virusafe" categoriseşte un program, sau părţi ale lui, ca fiind "suspecte de viruşi", el generează un director temporar cu "material de joacă". Ulterior software-ul suspect va fi implementat în acest director şi "Virusafe" urmăreşte ceea ce se întîmplă.

Dacă datele memorate în directorul de test vor fi alterate sau modificate, programul va şterge directorul de test împreună cu fişierul original din care provine programul suspect. Apoi virusul recunoscut va fi introdus automat în lista "programelor de sabotaj cunoscute". La următoarea încercare de infiltrare virusul va fi şters imediat.

Programul antivirus al celei de-a doua generaţii este imun la infectarea proprie cu vre-un virus. El constă din 6 segmente, fiecare segment conţinînd secţiuni de program reacoperibile. Dacă la autotestare unul dintre cele 6 segmente este depistat ca fiind "virusat", atunci această secţiune va fi ştearsă. Programul este în stare să se regenereze din elementele de program redundante existente în celelalte secţiuni.

(I.M.)

Dublaţi capacitatea discului Winchester !

În luna februarie 1991 firma Stac Electronics lansează pe piaţă un produs care se va bucura de un deosebit succes în lunile următoare: **STACKER 1.10.**

Ce este de fapt STACKER ? Nimic altceva decît un produs software de compactare a datelor în timp real.

În esenţă, STACKER este capabil să dubleze capacitatea discului Winchester fără probleme deosebite. Este remarcabil faptul că poate fi instalat sub diferite variante DOS ca: MS-DOS şi PC-DOS versiuni mai mari decît 3.0, DR-DOS 5.0, Compaq şi Tandon DOS.

Recunoaşte partiţiile mai mari de 32 MB, se instalează uşor şi nu este necesară reformatarea discului Winchester la instalare.

Există două modalităţi distincte de instalare şi anume:

- instalare incrementală, în acest caz toate datele de pe discul Winchester sînt compactate, dar înainte de aceasta se elimină toate "gap"-urile de pe disc, de aceea durata instalării este mai lungă.

- instalare numai pe zona liberă a discului, durata acestui tip de instalare fiind mai scurtă.

Viteza de prelucrare a datelor nu este diminuată de operaţiile de compactare/restaurare din RAM, dimpotrivă ea poate fi chiar accelerată luînd în considerare faptul că acum capetele de citire/scriere pentru aceeaşi cantitate de informaţie vor efectua numai jumătate din deplasarea anterior necesară.

La instalare STACKER creează o nouă unitate logică, aceasta se comportă ca şi cînd utilizatorul ar avea la dispoziţie un disc Winchester de capacitate dublă.

Odată instalat STACKER oferă patru noi comenzi specifice şi anume:

- **SDIR**, furnizează informaţii asupra raportului de compactare pentru fiecare fişier dintr-un subdirector.
- **SCHECK**, este similar cu comanda CHKDSK din DOS.
- **SCREATE**, permite crearea de noi unităţi logice de capacitate dublă.
- **SREMOVE**, anulează o unitate logică creată anterior.

Lansarea în execuţie este asigurată de un "device driver", înscris automat la instalare în fişierul CONFIG.SYS, care ocupă între 30 KB şi 50 KB din memoria operativă. Cu ajutorul unor programe de gestiune a memoriei precum QRAM, ABOVE DISK, QEMM 386 etc., se poate încărca acest "device driver" în memoria extinsă, eliminînd astfel şi acest mic neajuns referitor la memoria operativă.

Trebuie avut în vedere şi faptul că datele compactate cu Stacker nu mai pot fi citite decît cu ajutorul driver-ului specific. Dacă dintr-un motiv oarecare vă decideţi să nu mai utilizaţi produsul - pentru că, de exemplu, nu aveţi memorie extinsă şi doriţi să utilizaţi un program care necesită întreaga memorie, nepermiţîndu-vă alocarea a nici măcar 30 KB în alte scopuri - atunci fişierele compactate nu mai sînt vizibile decît după o decompactare cu driver-ul specific Stacker instalat.

În afara variantei software, firma Stac Electronics oferă şi o variantă hardware. Aceasta constă dintr-o plachetă care conţine un coprocesor specializat şi 32 kB memorie RAM, ceea ce măreşte şi mai mult viteza de lucru, cît şi raportul de compactare a datelor.

Făcînd un mic calcul, vom constata că achiziţionarea acestui produs este o investiţie rentabilă, chiar dacă vom converti numai un disc Winchester de 40 MB la 80 MB. Preţul unui nou disc Winchester împreună cu controller-ul aferent depăşeşte preţul STACKER-ului. Produsul poate fi achiziţionat de la firma AbMod, tel. 95-156350

(Publik Francisc, AbMod, Cluj-Napoca)

Expoziţie

Expoziţia organizată de firma OCTOGON COMPUTERS SRL din Tîrgu Mureş, în perioada 10-12 septembrie, în sala de expoziţii a casei de cultură a tineretului Mihai Eminescu, a stîrnit interesul general al celor interesaţi de tehnica de calcul. Vizitatorii au putut aprecia calitatea echipamentelor prezentate şi a aplicaţiilor realizate cu aceste echipamente. Cei interesaţi au putut vedea o staţie grafică de lucru pentru proiectare asistată de calculator compusă dintr-un sistem 386 cu coprocesor matematic Cyrix 83D87 de 33 MHz şi echipată cu un plotter Hewlett-Packard 7575A. Un alt punct de atracţie l-a constituit demonstrarea posibilităţilor de realizare de sisteme de achiziţii de date cu calculatoare DTK. Un stand a fost destinat prezentării de aplicaţii economice şi financiar-contabile realizate de colegi de la alte firme. Posibilităţile de lucru într-o reţea Novell ale unui program de evidenţă a furnizorilor şi comenzilor dintr-o întreprindere au fost demonstrate pe un stand constituit din opt calculatoare, legate într-o reţea de tip Arcnet. Prezentările au fost condimentate cu un set de programe de demonstraţii grafice, menite să pună în evidenţă calitatea excepţională a calculatoarelor DTK, calculatoare foarte apreciate şi în SUA, aşa cum a rezultat şi dintr-o anchetă realizată de reputata revistă PC Magazine, în luna iunie '90, în rîndul a 18.000 de utilizatori americani. Calculatoarele DTK au obţinut punctaje ridicate la toate întrebările referitoare la fiabilitate şi performanţe. Acesta este unul din motivele pentru care firma OCTOGON COMPUTERS SRL a hotărît importul acestor echipamente. Reputatele imprimante EPSON, dintre care au fost prezentate cele de uz general FX-1050, una color LQ-2550 şi una rapidă DFX-8000, s-au integrat perfect în configuraţiile prezentate şi au ilustrat în mod convingător dorinţa firmei OCTOGON COMPUTERS SRL de a oferi partenerilor echipamente de calitate, în configuraţii specifice aplicaţiilor beneficiarului. Pentru echipamentele livrate firma asigură garanţie şi service postgaranţie. OCTOGON COMPUTERS SRL asigură partenerilor şi software autorizat, fiind capabilă să satisfacă necesităţile beneficiarilor, conform dorinţelor acestora, începînd cu determinarea necesităţilor partenerilor, procurarea de echipamente, instalare, service, proiectare de aplicaţii şi şcolarizare.

Telefonul firmei este 954/38095.

Factorul 20%

Microsoft. Making it all make sense. Deviza unuia dintre cei mai mari producători de software nu este o vorbă goală. Aproape că nu este domeniu al soft-ului destinat compatibilelor IBM PC în care să nu existe cel puţin un produs purtînd prestigioasa semnătură. O succintă (şi desigur, incompletă) enumerare a acestora va fi, sperăm, edificatoare: MS-DOS (sistem de operare), LAN-Manager (sistem de operare în reţea), Windows (suprafaţă utilizator grafică), Multiplan, Excel (programe de calcul tabelar), Word, Word for Windows (editoare de texte), Works (sistem integrat), Project, Project for Windows (programe de administrare proiecte), Powerpoint (program de prezentare grafice), Windows Paintbrush (program de pictură şi desen), MS-Basic, MS-Fortran, MS-Cobol, MS-C (compilatoare), Windows SDK (programare Windows), SQL Server (server de baze de date). Completăm această enumerare cu sublinierea că cele mai multe dintre aceste produse se plasează pe primele locuri în topul preferinţelor utilizatorilor de compatibile IBM PC (v. nr. 4/91 al revistei "if", pag. 4) şi cu menţiunea că pentru Word, Excel, Powerpoint şi Works există versiuni Apple Macintosh care însă nu epuizează gama de produse Microsoft destinate acestor microcalculatoare.

Însă orice prezentare a firmei Microsoft este incompletă în lipsa conturării personalităţii aceluia care îi este fondator, preşedinte, director executiv, cel mai important acţionar şi, probabil, angajatul care munceşte cel mai mult: **William H. Gates**. Cel care este supranumit "Billion Dollar Gates" este considerat în momentul de faţă persoana cea mai influentă în industria calculatoarelor. În acest context, personalitatea sa a dat naştere unei game foarte largi de atitudini, de la admiraţie şi respect

Bill Gates

la suspiciune şi invidie (fenomenul este explicabil, dacă nu chiar normal). În rîndurile care urmează sînt cuprinse cîteva dintre cauzele şi dovezile acestei capacităţi de a influenţa direcţiile de dezvoltare ale industriei calculatoarelor.

Din cei 60-70 de miliardari din S.U.A. 5 aparţin industriei calculatoarelor iar 2 dintre aceştia din urmă, Bill Gates şi asociatul său, Paul Allen sînt de la Microsoft (ceilalţi trei sînt William Hewlett, David Packard şi H. Ross Perot de la EDS). Gates a devenit cel mai tînăr miliardar "self-made" din istorie în 1986, cînd Microsoft a devenit companie publică. Avea 30 de ani. Astăzi, averea sa este estimată la 2-3 miliarde dolari, aproape în întregime plasată în acţiuni Microsoft. Salariul şi celelalte drepturi cuvenite funcţiei de director executiv sînt, prin comparaţie (cu cele 2-3 miliarde, nu cu salariul dvs, evident), modeste: 207.000$ în 1989. Puternica personalitate a lui Gates a fost conturată în aproape 50 de editoriale de revistă în 1990. Publicaţii americane cum sînt: Computer Retail News, Electronic Business, Personal Computing şi Upside îl plasează cu regularitate printre cei mai influenţi factori executivi, deseori considerîndu-l cel mai influent. PC Magazine i-a decernat în 1986, deci la numai 30 de ani, al său Lifetime Achievement Award, lucru ce poate părea surprinzător dacă ne gîndim că premiile de acest gen semnifică aprecierea şi recunoaşterea unei activităţi de o viaţă. Dacă însă avem în vedere remarca unui colaborator al lui Gates din perioada de început a industriei calculatoarelor personale, potrivit căreia atunci "un an era o viaţă" faptul apare explicabil. După cum explicabilă este (mai mult pentru ei decît pentru mulţi dintre noi) apariţia, prevăzută pentru 1992, a unei biografii cuprinzătoare (500 pagini) a celui care în anul apariţiei va împlini 36 de ani.

Şi dacă tot am ajuns la biografie, să punctăm rapid ascensiunea lui William Gates: absolvent al Universităţii Harvard, elaborează - împreună cu Paul Allen şi Marty Davidoff - un Basic pentru unul dintre primele tipuri de calculatoare personale, MITS Altair, după care se mută - împreună cu Allen - la Albuquerque pentru a continua dezvoltarea de software pentru MITS. Revenit acasă la Seattle reuşeşte să obţină pentru mica sa firmă, MicroSoft (aşa se scria atunci), contractul pentru elaborarea sistemului de operare pentru PC-urile gigantului IBM, devansînd firma Digital Research. Cumpără apoi de la Tim Paterson sistemul de operare QDOS (Quick-and-Dirty (?) Operating System) pentru 50.000$, pe care îl va folosi ca bază pentru MS-DOS.

Acesta a fost începutul. Astăzi Microsoft are 5-6 mii de angajaţi şi vînzări al căror volum depăşeşte un miliard de dolari anual. Este una dintre cele mai profitabile companii din industria de calculatoare, dacă nu cea mai profitabilă. O rată de creştere de 31% şi un profit de 22,3% pentru acţionarii Microsoft în ultimii 5 ani sînt numai două dintre argumentele care vin în sprijinul acestei afirmaţii.

(continuare în pag. 9)

Strategia Microsoft

Strategia de viitor a firmei Microsoft, la capitolul sisteme de operare (S.O.), ar putea fi rezumată astfel: Windows pe toate platformele hard. Acesta este răspunsul celor de la Microsoft la campania de promovare a sistemului de operare OS/2, declanşată de IBM.

Încă din 1985, IBM şi Microsoft îşi făcuseră planuri comune de promovare a S.O. OS/2, ca urmaş al S.O. MS-/PC-DOS. După mulţi ani de speculaţii, de teste soft şi de discuţii se pare că drumurile celor doi uriaşi se vor despărţi. În ultimul timp ei vorbesc mai des unul despre altul, decît unul cu altul. La o expoziţie de PC-uri organizată la New York, IBM a arătat, mai mult pe sub mînă, software care trebuia să demonstreze cît de uşor poate provoca Windows căderea sistemului. Numele programului, după cum afirmă apropiaţii casei, este "Bad App". Pe de altă parte, la o întrunire cu analişti ai pieţei, la sfîrşitul lunii iulie, Microsoft a prezentat programul "Terminator", care demonstrează cît de instabil este OS/2. Atît Bad App cît şi Terminator au fost realizate pentru necesităţi proprii şi nu pentru a fi comercializate. A vorbi despre un divorţ între cei doi parteneri este, totuşi, prea devreme. Atît doar că drumurile lor se despart. Momentan pot fi reţinute, ca fapte, şi respectiv declaraţii de intenţie, trei aspecte: mai întîi că nu va exista un sistem de operare cu numele OS/2 3.0 produs de Microsoft. Locul acestuia va fi luat de Windows NT (New Technology). În al doilea rînd Windows va fi astfel dezvoltat încît să poată fi utilizat pe o paletă hard cît mai largă. În acest scop vor exista două linii Windows: Windows X.X şi Windows NT (New Technology). În al treilea rînd, imediat după terminarea lui OS/2 2.0, IBM şi Microsoft îl vor oferi spre vînzare pe căile obişnuite, Microsoft prin intermediul OEM, deci al producătorilor hard care au cumpărat licenţa, şi IBM prin intermediul canalelor proprii de comercializare. Strategia Microsoft, revăzută, va funcţiona deci abia pentru urmaşii lui OS/2 2.0.

Microsoft se vede obligat să-şi revizuiască strategia, în privinţa sistemelor de operare, ca urmare a "impresionantului succes" pe care l-a avut Windows 3.0. Momentan, în lume, există cca. 4 milioane de instalări Windows 3.0 la cca. 70 de milioane de PC-uri DOS.

Windows va fi acceptat ca standard. După datele furnizate de institutul de prospectare a pieţii IDC, în ultimul an, în Germania, au fost vîndute 1,5 milioane PC-uri cu procesoare Intel 80286/80386 şi 80486 şi cca. 300.000 de copii Windows. În 1991 se va ajunge la 600.000 de copii Windows. După cum afirmă specialiştii, pînă la sfîrşitul acestui an pe o cincime din PC-uri se va lucra sub Windows. IDC apreciază că pînă în 1995 în Germania vor exista 3 milioane de instalări Windows pe PC-uri şi că Windows va fi utilizat în proporţie de 40-50% pe piaţa profesională şi neprofesională. Încă din 1993 80% din sistemele noi vor fi livrate cu Windows.

Windows va deveni un standard, deoarece suprafaţa utilizator grafică va fi acceptată de mulţi producători, proiectanţi şi utilizatori. Cu toate că la sfîrşitul lui 1990 Windows 3.0 era pe piaţă de abia 8 luni, suprafaţa utilizator grafică a lui utilizată în aplicaţii a adus o cifră de afaceri de 400 de milioane dolari. Aplicaţiile Macintosh au atins o cifră de afaceri de 463 milioane dolari. Diferenţa între cifrele de afaceri realizate cu aplicaţii orientate caracter şi aplicaţiile grafice este încă semnificativă. În cazul livrărilor de produse noi balanţa se va înclina, însă, în favoarea aplicaţiilor grafice. Această tendinţă este sprijinită de uneltele din domeniul Enduser-Programming (unelte pentru programatorii de aplicaţii), cum ar fi de exemplu, Object Vision, Oracle Card, Toolbook, Object Script sau Visual Basic.

În timp ce Windows şi aplicaţiile realizate cu acesta pot da impulsuri semnificative pieţei, după Microsoft, OS/2 nu a avut nici pînă astăzi ecoul aşteptat. La sfîrşitu! lui 1990 existau cca. 600.000 de instalări OS/2. Această situaţie s-ar putea schimba odată cu livrarea, la sfîrşitul acestui an, a lui OS/2 2.0. Succesul lui OS/2 va depinde de calitatea sa şi de respectarea promisiunii de a se putea rula sub el fără probleme, şi fără modificări, aplicaţiile DOS şi Windows. În acest caz n-ar mai fi necesar ca Microsoft să aducă pe piaţă aplicaţii cu PM (Presentation Manager). Teama că aplicaţiile Windows ar putea într-adevăr să ruleze mai bine sub OS/2, a fost un alt motiv care a determinat Microsoft să-şi revadă strategia de viitor la capitolul sisteme de operare. Dacă ceea ce promite IBM se va întîmpla, va fi rău pentru piaţa Windows, dar va fi bine pentru Windows - API (Application Program Interface).

Important este, de asemenea, dacă IBM-ului îi va reuşi să cucereasc ă proiectanţii de soft pentru OS/2, pentru a scrie aplicaţii specifice OS/2. Deoarece pentru ca OS/2 să se poziţioneze mai bine decît Windows, vor trebui să existe suficiente aplicaţii atractive pe piaţă. În caz contrar doar avantajele care vor decurge din faptul că aplicaţiile Windows vor putea fi rulate mai rapid sub OS/2 şi că va fi posibil un multitasking real, nu vor fi suficiente pentru a determina o "masă critică" de utilizatori să lucreze cu OS/2. Din aceste motive este de înţeles interesul pe care-l are Microsoft ca Windows să poată concura OS/2.

IBM promovează S.O. OS/2 pe piaţă cu toată puterea. În SUA, cine schimbă vechiul PC-DOS cu OS/2 1.3 nu trebuie să plătească decît

99$. Atunci cînd va fi disponibil OS/2 2.0, actualizarea va fi făcută gratuit. IBM duce o agresivă, dar acceptabilă din punctul de vedere al utilizatorilor, campanie de preţuri pentru a acapara pentru OS/2 o parte din sectorul de piaţă deţinut de Windows.

Conflictele de interese dintre IBM şi Microsoft, în ceea ce priveşte Windows şi OS/2 nu pot fi negate. Dacă situaţia se va schimba după anunţarea noii strategii Microsoft, rămîne de văzut, deoarece abia dezvoltarea în continuare a S.O. OS/2 2.0 este afectată. Cu Windows, sau cu OS/2 2.0 pot fi utilizate cca. 90% din aplicaţiile de pe PC-uri. Numai pentru restul de 10% va fi necesar OS/2 sau Windows NT ca sistem de operare server, pentru High-End-Multitasking, pentru comunicaţii PC-Host etc.

Atît IBM cît şi Microsoft sînt interesate în crearea unui S.O. portabil care să fie privit ca o dezvoltare tehnologică a lui OS/2 2.0. Acest proiect ar putea fi susţinut şi de IBM cu toate că pînă acum nu s-au făcut confirmări oficiale. În această direcţie se îndreaptă acordurile încheiate între IBM şi Borland. crearea unui sistem de operare portabil, inclusiv pentru procesoarele RISC. Cît de mult vor fi afectate relaţiile de cooperare de viitor, între IBM şi Microsoft, nu se poate şti încă.

Controversat este modul de introducere pe piaţă a S.O. OS/2. Microsoft doreşte ca migraţia să se realizeze via Windows, în timp ce IBM ar dori să înlocuiască imediat S.O.DOS cu OS/2. O premisă importantă, pentru Microsoft, este ca noile produse să rămînă compatibile cu cele deja existente pe piaţă, pentru a putea fi utilizate în continuare produsele instalate. Din acest motiv vor exista două linii Windows: Windows X.X. şi Windows NT. Windows X.X. este gîndit pentru marea majoritate a aplicaţiilor PC. Suplimentar va fi disponibil Windows cu "extensii.Pen" pentru utilizare de Notepads. Se are în vedere şi includerea HDTV (High Definition TV), controlul şi comanda copiatoarelor, fax-urilor, telefoanelor şi a altor mijloace de muncă şi organizare din birouri după cum a afirmat Bill Gates, la o conferinţă de presă în Redmond, USA, la sfîrşitul lunii iulie. Este de imaginat şi o deschidere spre jocurile video, pentru a putea satisface astfel întreaga piaţă de bunuri de consum tradiţionale. Deviza Microsoft în prezent: Windows everywhere, ca sistem scalabil, inclusiv multimedia (integrare de imagini, vorbire şi date), Pen-Technology şi orientare obiect, într-o perspectivă de timp medie. Baza o vor constitui în continuare Windows-APIs (Application Program Interface).

Windows NT, dimpotrivă, mizează pe un nou sistem de operare, de aici şi denumirea New Technology şi va fi un sistem de operare High-End (de vîrf) pentru servere şi aplicaţii complexe pe staţii de lucru. După datele furnizate de Microsoft, Windows NT va fi scris în C şi va fi astfel portabil şi pentru alte platforme hard. Încă din acest an vor fi disponibile SDK-uri (Software Developement Kits) pentru Windows NT. Pe un produs finit se poate conta abia pentru anul viitor.

Windows NT corespunde astfel cu ceea ce Microsoft desemnase a fi OS/2 3.0 şi care urma să fie unul din cele două sisteme de operare de promovat în cadrul iniţiativei ACE (Advanced Computing Environment). La conferinţa de presă de bilanţ a firmei Microsoft, de la sfîrşitul lunii iulie, din Redmond, SUA, Bill Gates nu a respins categoric posibilitatea portabilităţii şi pe alte procesoare RISC, de exemplu pe procesoarele SPARC.

Din cauza conflictului de interese apărut între IBM şi Microsoft se pune întrebarea: într-o intreprindere, în viitorul apropiat, trebuie să se mizeze pe Windows sau pe OS/2 ? Interesant va fi pentru utilizatorii care folosesc în prezent OS/2 1.X într-o versiune OEM provenită de la Microsoft şi care vor dori să treacă apoi la Windows NT. Pentru aceştia va trebui să existe un program de conversie (Migration Layer) care să le dea posibilitatea de a converti aplicaţiile create cu Presentation Manager pentru Windows NT. Sarcina lui Migration Layer: conversia PM-API în Windows-API.

(R.M.)

Factorul 20%

(continuare din pag. 7)

În ceea ce priveşte influenţa firmei, cele peste 60 milioane de calculatoare pe care rulează software Microsoft reprezintă numai partea vizibilă a acesteia. Cercetările firmei în domeniile multimedia, comunicarea interaplicaţii, interfaţa utilizator pentru "creionul electronic" (pen based computing), standarde CD-ROM, dovedesc că Microsoft, ca orice firmă care are de apărat o poziţie conducătoare, priveşte în perspectivă.

Revenind la pesonalitatea lui William Gates, ceea ce ni se pare a fi lucrul cel mai important este ponderea acestuia în cadrul firmei, apreciată la 20%. Cînd Gates este considerat "factorul 20%" nu se face referire la volumul acţiunilor sale, acesta ridicîndu-se, de fapt, la 38% ci reprezintă o evaluare a importanţei acestuia pentru firmă. Înseamnă că din orice se face sau se poate face la Microsoft, o cincime este sau va fi realizată de un singur om. Ceea ce, veţi recunoaşte, este extraordinar!

(Răzvan Grigorescu, Octogon, Tîrgu Mureş)

Protecţia programelor

Avem actualmente lipsuri de tot felul, nu ne putem plînge! Dar ştim bine că resursele sînt nu numai materiale, umane şi financiare, ci şi informaţionale (neglijate pînă acum la noi). Sub acest aspect, există la noi un imens vid de legislaţie, o mare lacună în cultura juridică. Între domeniile aflate în umbră, este şi protecţia intelectuală. Ceva interesant s-a dicutat pe această temă la REP '90 (Ediţia a XI-a), 7- 9 sept. la Snagov. Fiind vorba de consemnări de discuţii, textul este în mod inerent incomplet.

S-a vorbit desigur despre etapa actuală: Cel mai mare pericol este provizoratul. Se ştie că provizoratul tinde să devină definitiv (starea de provizorat este cea mai sigură). Riscul cel mai mare este să prelungim prea mult această tranziţie... Despre **tranziţia plan-piaţă**: Fenomenul este greşit înţeles, dacă primul pas se referă la micul întreprinzător. Dacă se începe cu această atomizare, risipire, dispersare a forţelor, necuplată cu o demonopolizare a proprietăţii de stat, înseamnă că ne reîntoarcem la perioada de dinainte de '40. Fără o piaţă liberă, nu poate fi vorba de o libertate reală, nici în economie şi nici în general...

În condiţiile trecerii de la plan la piaţa liberă, creşte rolul mijloacelor de protecţie intelectuală şi, concret, de protecţie industrială. Schimbarea de concepţie la nivel general implică şi o schimbare în legislaţie. Există preocupări, îngrijorări. Legislaţia este în curs de revizuire, s- a făcut un proiect de lege a invenţiilor. A fost publicat, dar ecourile au fost foarte slabe.

În 1985, pe plan mondial, s-a încercat introducerea unui sistem de protecţie pentru plachete şi circuite integrate. Cei mari, SUA, Japonia au realizat legislaţia, cei mici au vrut să scape, au subminat, au boicotat proiectul. Americanii le-au dat ultimatum: altfel nu le mai vînd cipuri. În '87- 88 toţi din CEE au trebuit sa accepte protecţia, să adopte legi naţionale, dar fiecare cum vroia. În 1988 s-a simţit nevoia unui sistem unificat de protecţie. În 1989 (26 mai, la Washington) s-a semnat Tratatul internaţional asupra protecţiei industriale asupra CI. În consecinţă, a fost trimis spre consultare şi la noi în ţară un proiect de lege. De la OSIM s-a ocupat de această problemă V. Erhan, dar se pare că cei interesaţi n-au luat la cunoştinţă.

- Din 100 de invenţii, 5- 6 sînt foarte valoroase, ar putea să ne scoată din substanţa aia în care eram... Dar inventatorii s-au lovit de..., aşa cum s-a spus, invidia aceea perversă, nici măcar conducerea proprie nu i-a sprijinit. Din cauza necinstei.
- Ai un produs brevetat - cine te împiedică să-l faci şi să-l dai la export? În momentul de faţă, cine verifică dacă nu foloseşte altul brevetul tău? Legea actuală este făcută pentru economia socialistă, deci cedezi titlul întreprinderii, poate fi folosit de toţi. Nu-i avantajos şi nici stimulativ: întreprinderea titulară plăteşte taxe, n-are nici un avantaj. Dacă titlul ar fi personal, respectivul ar urmări aplicarea...

Protectia programelor de calculator

Şi aici, americanii au impus protecţia. Nu există încă un tratat internaţional, dar avem exemple concrete. Cehoslovacia, pentru a obţine clauza naţiunii celei mai favorizate, era condiţionată (între multe altele) şi de rezolvarea protecţiei programelor de calculator prin legea drepturilor de autor.

La OSIM, de problema aceasta se ocupa Dan Şova şi Doina Manicatide.

- Cum se poate face protecţia produselor software?

- În nici un fel! Legislaţia actuală din ţară **NU** poate asigura protecţia programelor. Nu se poate aplica legislaţia privind protecţia invenţiilor. Pe plan mondial, cei mari încearcă să forţeze protecţia produselor de informatică.

Literatura privitoare la brevetabilitate afirmă: *NU se poate. Softul nu constituie o soluţie tehnică în accepţiunea brevetabilităţii.* O soluţie tehnică brevetabilă se referă la aspecte concrete ale tehnicii, iar în programare se lucrează cu abstractizări. Sînt brevetabile soluţiile tehnice concrete, palpabile fizic. În general nu sînt brevetabile **noile aplicaţii**, ci revrendicarea lor.

De altfel, invenţiile pionier sînt foarte rare, mult mai frecvenţi sînt paşii tehnici de perfecţionare. Remediile, ameliorarea unei soluţii cunoscute. De aceea, în revendicare se descriu părţile cunoscute ("prior art") + elementele noi; de exemplu perfecţionarea unei scheme bloc.

O nouă aplicaţie pentru o schemă bloc cunoscută NU se brevetează: să presupunem că avînd un calculator pentru jucat şah, iau Epromul, îl reprogramez şi îl folosesc pentru ghidat rachete... nu pot să protejez noua aplicaţie!

La soft, nu există o nouă soluţie tehnică concretă, există doar o nouă utilizare, care s-ar putea breveta dacă duce la efecte deosebite pentru societate, dacă se fac cheltuieli foarte mari pentru programare. Dar utilizatorii nu vor, obstrucţionează.

Există posibilitatea protecţiei prin drepturi de autor. Primele decizii ale tribunalelor NU s-au referit la programele propriu-zise, ci la calculatoare şi programe. De exemplu: avem un desen tehnic, îl putem considera o "operă de artă", la fel o melodie, un film de desene animate...

Au fost programe brevetate, adică li se conferea recunoaşterea priorităţii internaţionale.

În 1990, ţara noastră a fost (?) la Geneva pentru protecţia pe baza drepturilor de autor. În CAER au fost discuţii, orientate spre *cum să NU se adopte*.

Sînt protejate programele care reflectă un proces tehnologic, deci în care se regăsesc paşii de proces respectivi - se poate considera că se descrie o metodă. Din tot programul, elementul de protecţie este revendicarea, dacă nu descrii programul în întregime, cu toţi cei 10.000 de paşi pe care-i are, înseamnă că nu este protejat.

Această situaţie nu-i acceptată nicăieri în lume. Din '85 s-au făcut presupuneri (?), modificări în legislaţia privind drepturile de autor: Japonia, RFG. Concret, nu s-a realizat mare lucru, cu excepţia legii japoneze, care cere precizări amănunţite privind programul. Solicitantul trebuie să depună o cerere şi programul este înscris într-un registru în care sînt înregistrate drepturile de autor.

În privinţa drepturilor de autor, exista două concepţii principale:

- a) cea continentală: acordul semnat la Berna în 1890 prevede protecţia operelor literare, artistice şi ştiinţifice fără înregistrare. Durata = viaţa autorului + 50 de ani pentru moştenitori. Este limitată în timp şi spaţiu. Timp: 15 ani la noi, în alte ţări 8, 12 sau 20 de ani. Este în studiu reglementarea armonizării legislaţiei - să se standardizeze la 20 de ani (sîrbii aveau 8 ani). Dupa această perioadă, operele intră în patrimoniul public, se pot consulta şi folosi gratuit. Sînt în studiu următoarele aspecte:

» - cum se face înscrierea;
» - ce aplicaţii foloseşte(?);
» - cum se justifică.

Bush a acordat clauza Cehoslovaciei, deci au acceptat protecţia programelor!

În consecinţă, concepţia continentală prevede nu înregistrarea, ci recunoaşterea drepturilor de autor pentru o operă, din momentul în care a fost realizată.

- b) cea engleză, cu Copyright. Este o noţiune puţin deosebită: nu înseamnă recunoaşterea automată a oricăror realizări, dacă o realizare nu a fost înregistrată, nu este recunoscută. Trebuie să se facă înregistrarea în toate ţările în care se doreşte ca opera să fie protejată. Deci, dacă un program n-a fost înregistrat în ţară la noi, nu este protejat.

- Noi vrem să facem programe şi să le protejăm, dar le facem utilizînd un soft de bază. De unde-l avem? Nu este şi el protejat?

În '85 s-a început lucrul la protecţia circuitelor integrate, în '87, sub presiunea americanilor şi japonezilor, au fost adoptate 7 legi naţionale în Europa de West. În '89 a fost semnat Tratatul de protecţie pentru circuitele integrate. A fost semnat şi de ţara noastră, astfel că de acum înainte utilizarea unui circuit integrat copiat într-un aparat (de larg consum, tehnică de calcul, etc) poate crea grave probleme dacă aparatul, chiar brevetat, ajunge în West.

Clauze privind înregistrarea circuitelor integrate: nu mai tîrziu de 2 ani de la utilizarea lui comercială. Se brevetează arhitectura, structura, măştile. N-are nimeni dreptul să zică: *"N-am ştiut!"*

Situaţia este similară şi pentru soft. Trebuie stabilit clar UNDE, cine păstrează, care-s condiţiile de păstrare.

Se fac excepţii:

» - în scop de cercetare se poate analiza o lucrare, pentru ca să nu se cadă din întîmplare peste aceeaşi idee;
» - pentru învăţămînt.

Trebuie să cunoaştem toate portiţele!

Trebuie să existe o societate pentru protecţia drepturilor respective, aşa cum sînt Uniunea Scriitorilor sau Uniunea Compozitorilor, să aibă un birou care să se ocupe de verificarea respectării drepturilor de autor.

Deci, după adoptarea legislaţiei respective, va trebui stabilit un cadru executiv corespunzător, o **Poliţie de Soft**!

De exemplu, pentru muzică: compozitorii au un salariat care ascultă toate posturile de radio, notează că "bucata X a lui Cacoveanu a fost cîntată ieri 25" pe postul Bucureşti III", le percepe taxele corespunzătoare, iar la sfîrşitul anului îi spune lui Cacoveanu că are de luat drepturi de autor de 28.000 lei, din care se scade taxa de urmărire de 10%.

Aşadar, trebuie ceva similar şi pentru programe. Depozitar: ceva gen BNP. A depozita legal înseamnă să ţii sigilat, răspunderea este foarte mare, este o problemă de încălcare a unor drepturi. Cum se aduce la cunoştinţă unui potenţial beneficiar? Trebuie spus că programul respectiv a fost înregistrat, la ce poate fi folosit... trebuie făcută publicitate, reclamă (să se sublinieze avantajele: viteză, precizie, economie de memorie) pentru ca beneficiarul să cumpere sau să închirieze programul.

Sînt tot felul de probleme

» - biblioteca are drept de copiere?
» - folosim un limbaj de programare. În Japonia, acestea sînt excluse de la protecţie!
» - cum se poate verifica folosirea ilegală a unui program?
» - intră şi mediile de programare în această categorie? Dacă fac o aplicaţie în Data Base şi vreau s-o vînd. Trebuie să vînd şi dBase-ul beneficiarului? Trebuie să-i spun cumpărătorului sub ce mediu a fost elaborată aplicaţia?
» - daca vreau să export în West un program făcut cu DataBase IV, care-i furat, intru în puşcărie? Nu, utilizatorul trebuie să-l cumpere?

Nu sîntem noi cei mai mari hoţi. Măcar să fi fost, am fi acum precum Hong-Kong-ul sau Coreea de Sud! Mai este în vigoare decretul din '63, pe care nu-l respectă nimeni, nu ţine cont de legislaţia internaţională...

(Cristian Malide, ICCE)

L40 SX - una dintre cele mai frumoase surprize ale firmei IBM

După numeroase tentative nereuşite, Big Blue a apărut cu un nou model, un laptop autonom care confirmă revenirea numărului unu pe piaţa explozivă a calculatoarelor portabile.

În 1990 s-au vîndut doar 10.000 de laptop-uri autonome avînd la bază microprocesorul 80386SX, astfel că apariţia lui L405X, cu funcţionalităţi standard, pe care concurenţii le oferă opţional, va declanşa o luptă acerbă între producători.

L40 dispune de un microprocesor 80386SX cu frecvenţa de ceas de 20 MHz. În versiunea de bază el conţine un modem/fax al firmei Com 1 livrat într-un Kit complet hard-soft. El permite transferul de fişiere la viteza de 2400 bds (V23, V21, V22 şi V22 bis).

Protocoalele MNY şi S de corectare a erorilor şi de compresie a datelor sînt integrate, de asemenea un modul V27 ter/29 care permite accesul la Fax Groupe III. Light Manager al lui Com 1 funcţionează sub MS DOS şi poate deveni în acest caz rezident. Suportă de asemenea "en tâche de fond" emisia în timpul recepţiei fax. Telecopiile pot fi transmise imediat după prelucrarea cu procesorul de texte Evolution (IBM). Memoria internă este de 2 Mo (80 ns) extensibilă prin două slot-uri pînă la 18 Mo.

Afişarea este făcută pe un ecran LCD STN de 10", spre deosebire de celelalte portabile care au doar 8" sau 9" Ea permite modul VGA cu 32 niveluri de gri. Ca la toate ecranele LCD pasive este imposibilă evitarea efectului "fantomă" în timpul deplasării cursorului. IBM a găsit însă un subterfugiu: imaginea pointerului se estompează în timpul mişcării, această funcţiune fiind parametrizabilă.

Harddisk-ul de 2 1/2" al firmei Conner Peripherals are o capacitate de 60 Mo cu timpul de acces de 19 ms. Lectorul de dischete este de 3 1/2" 1,44 Mo. Probabil lectorul de 2,88 Mo care este aşteptat şi care va funcţiona sub DOS 5.0, va fi opţional în viitoarea configuraţie.

Referitor la tastatură, IBM nu a acceptat nici un compromis privind talia tastelor şi amplasarea lor. Doar tastele de deplasare a cursorului au fost mutate puţin mai la stînga. Tastatura numerică separată este furnizată în standard ceea ce constituie o premieră în domeniul portabilelor. Ea se poate conecta la portul rezervat mouse-ului. Aceasta nu înseamnă că utilizatorii sub Windows nu îl pot folosi, deoarece acesta se poate conecta într-o priză existentă în spatele modulului ce conţine tasstatura numerică. De asemenea există: o ieşire paralelă, una serială şi una pentru un ecran VGA extern. Utilizatorii pot alege două carcase pentru plăci externe conform standardelor ISA şi MCA.

Originală este existenţa pe panoul frontal a unor diode LCD care pot indica permanent: starea de încărcare a bateriei, accesul la harddisk şi la lectorul de dischete, lucrul cu modemul, funcţionarea difuzorului incorporat, starea de defilare a ecranului, modul pauză şi modul de lucru cu tastele numerice şi majuscule. Există deasemenea o diodă de alarmă în cazul depăşirii temperaturii sau umidităţii. La nivelul acestor diode este un comutator care permite gestionarea energiei. Viteza variază, în mod automat, în funcţie de încărcarea sistemului (5,10 sau 20 MHz). Acest mod permite reducerea cantităţii de energie utilizată prelungind astfel viaţa bateriei.

L40SX dispune de o baterie NiCd care în mod normal (la 20 MHz) poate funcţiona trei ore; cu ajutorul sistemului de gestionare a energiei durata de funcţionare poate ajunge la opt ore. În timpul înlocuirii acumulatorului o baterie tampon furnizează energia necesară calculatorului.

În sfîrşit, trebuie să menţionăm că portabilul este livrat fără nici un sistem de operare, astfel că utilizatorii pot face o alegere între DOS şi OS/2. Ce se mai poate aştepta în anul viitor de la IBM? Probabil un frate sau un văr sau de ce nu un notebook?

(Voicu Manole)

Tabel comparativ (Laptop 386SX - 20 MHz)

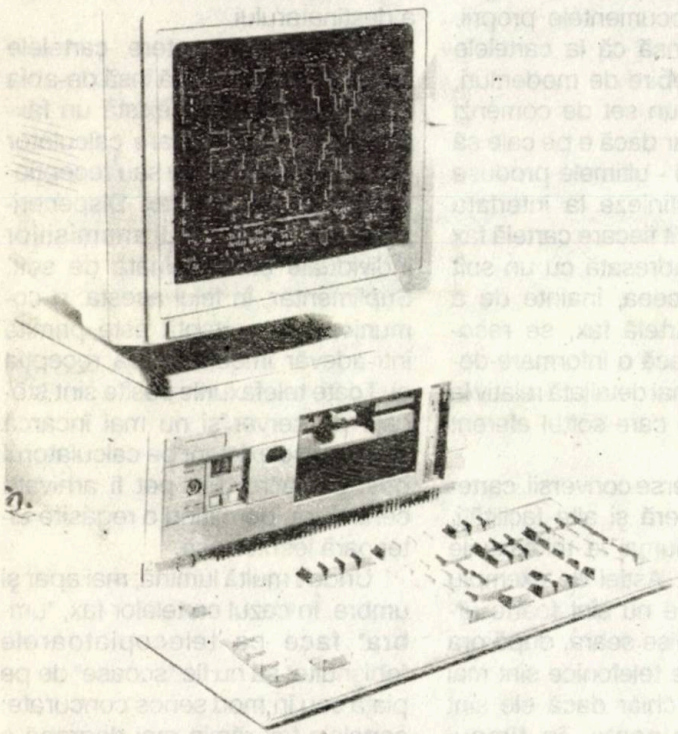
	AST	COMPAQ	DELL	GRID	**IBM**	TOSHIBA
Model	Premium Exec	STL 386 S/20	320N	1550SX	L40SX	T3200SX
Memorie standard	2 Mo	2 Mo	1 Mo	2 Mo	2 Mo	1 Mo
Memorie maximă	8 Mo	10 Mo	5 Mo	8 Mo	18 Mo	13 Mo
Harddisk	60 Mo	60 Mo	40/60 Mo	60/120 Mo	60 Mo	120 Mo
Lector de dischete 3.5" (1.44 Mo)	da	da	da	da	da	da
Ecran	LCD 32n*	LCD 16n	LCD 16n	LCD 32n	LCD 32n	plasmă
Port video VGA	da	da	da	da	da	da
Port serial	1	1	1	2	1 (2 opţ.)	2
Port paralel	1	1	1	1	1	1
Port mouse	da	da	da	nu	da	nu
Modem intern	nu	nu	nu	nu	da	nu
Tastatură numerică separată	opţional	opţional	opţional	opţional	da	opţional
Carcasă de extensie	nu	da (opţ.)	nu	da (opţ.)	da (opţ.)	nu
Autonomie	3 h	3 h	4 h	3 h	3 h	3 h 30'
Greutate	3 kg	3.4 kg	2.9 kg	5.4 kg	3.5 kg	1.9 kg
Preţ	4050$	7000$	3250$	8100$	6000$	6015$

(* nuanţe)

PC sau telefax?

- Cartele fax -

Ca alternativă la telecopiator ca echipament separat, PC-ul - dotat cu hard şi soft corespunzător - cîştigă din ce în ce mai mult teren ca emiţător şi receptor de telecopii. "if" vă prezintă avantajele şi dezavantajele cartelelor fax.



Avantajele faxului faţă de corespondenţa obişnuită sînt evidente. Telefaxul nu este numai una din cele mai rapide variante de a trimite texte, grafice, fotografii, scrisori, etc. oriunde în lume (teoretic), ci, deseori, o variantă chiar mai ieftină decît corespondenţa obişnuită. O scrisoare către un partener de afaceri din acelaşi oraş ajunge prin fax în numai cîteva minute şi costă, cu aproximaţie, 1 leu. Cu poşta normală, ar ajunge (cel devreme) a doua zi şi ar costa 4 lei.

Pentru a fi transmis ca fax, originalul este "citit" (scannat) de telecopiator, transpus într-un format propriu şi apoi trimis pe linia telefonică. La celălalt capăt al liniei, echipamentul receptor transformă informaţiile sosite într-o copie a paginii originale, pe care apoi o tipăreşte la o imprimantă. Astfel, fiecare participant primeşte o copie fidelă a originalului. De aici şi numele: "fax" provine de la "facsimil", termen folosit mai ales în industria tipografică (cf. DEX: "reproducere exactă a unui text, a unei semnături, a unui desen sau a unei picturi cu ajutorul fotografierii, al fototipiei, prin copiere manuală etc.")

Pentru ca telecopiatoarele din toate ţările să se poată înţelege între ele, în anii '70 s-a făcut normarea pe plan mondial a serviciilor telefax. Echipamentele au fost împărţite în 4 grupe (vezi caseta).

Pe cît de convingătoare sînt avantajele telefaxului - pentru utilizatorul unui PC, transmisia prin telefax înseamnă un procedeu mai degrabă anevoios. Înainte de a putea trimite o comunicare prin fax, ea trebuie să existe pe hîrtie, deci trebuie întîi tipărită (imprimată). Apoi hîrtia trebuie dusă la telecopiator - de regulă, în alt birou sau chiar în altă clădire. De-abia după ce imprimatul a fost introdus în telecopiator şi s-a format numărul corect al destinatarului, începe transmisia propriu-zisă.

Aceste dificultăţi pot fi evitate (aproape) fără probleme, folosind telecopierea prin intermediul PC-ului. Pentru a transforma un PC într-un telefax, este nevoie doar de o cartelă fax corespunzătoare, softul aferent şi - desigur - o linie telefonică la care să se facă conexiunea. Desigur, este nevoie - teoretic şi de aprobarea poştei.

În prezent, în Germania sînt instalate peste 700.000 de telecopiatoare; se aşteaptă ca la sfîrşitul acestui an să se depăşească numărul de 1.000.000. În lume, la începutul anului 1992 vor fi instalate peste 12.000.000 telefaxuri. La noi, extrapolînd numărul de faxuri instalate în judeţul Mureş - 100 - se poate aproxima la 6.000 numărul de telefaxuri instalate în ţară.

Ce spune poşta

La noi, cartelele fax încă nu au ajuns să fie cunoscute de ministerul sau forul de resort. Sau, cel puţin, nu şi-a făcut în nici un fel publică ştiinţa.

În prezent, sînt autorizate telecopiatoare standard (lista am publicat-o pe coperta 3 a numărului 6 din "if") şi modemuri standard (lista poate fi consultată în "if" nr. 7). Desigur, aceste liste se actualizează în permanenţă; amănunte la zi ar trebui să poată dea oricare poştă judeţeană (filiala RomPres Telecom) sau, respectiv, Ministerul Comunicaţiilor, Direcţia Generală de Reglementări, Bucureşti (această adresă pe un plic este suficientă ca să ajungă la destinaţie).

Taxa de instalare - dacă postul telefonic există - este de 1000 lei pentru un telecopiator şi de 1800 lei pentru un modem (2200 dacă e vorba de o instituţie). Abonamentul lunar este de 200 lei/lună pentru un telecopiator, respectiv de 800 lei/lună pentru un modem.

În afară de abonament, se mai plătesc, desigur, şi impulsurile consumate - la tariful obişnuit, 60 bani/impuls (toate valorile amintite sînt orientative - ele au fost adevărate pe data de 11 septembrie 1991, ora 12. Pentru valorile momentului, vă rugăm să vă adresaţi poştei locale, dacă sînteţi interesat).

Vis-a-vis de cartelele fax, cartelele modem, cartelele faxmodem şi alte astfel de bîzdîcuri, poşta noastră pare a avea poziţia tipică a ardeleanului faţă cu girafa. A obţine o autorizaţie pentru aşa ceva este în mod cert - încă, şi, sperăm, nu pentru mult timp - mult mai greu decît a pune în lucru respectivul bîzdîc. Cum încă şi amenzile sînt neclare, nu rămîne de făcut decît calculul tipic al studentului-făr-de-bilet: care e timpul după care nu mai contează amenda, "scorul" fiind oricum pozitiv ?...

Odată cu instalarea cartelei fax în PC, problemele mai sus amintite dispar: texte sau grafice pot fi concepute şi transmise direct din PC, fără să fie necesară imprimarea prealabilă a documentului ce urmează a fi transmis. Importante şi interesante sînt posibilităţile noi pe care le deschide un PC cu cartelă fax. Cînd un PC recepţionează un fax, atunci acesta e disponibil într-un format grafic propriu. Cu un soft adecvat - cel puţin teoretic - un fax ce conţine numai text ar putea fi convertit în fişier ASCII şi folosit în procesorul de texte cu care se lucrează în mod curent. Deasemenea, graficele ar putea fi convertite în diverse formate pentru a fi înglobate apoi în documentele proprii. Trebuie spus însă că la cartelele fax, spre deosebire de modemuri, nu există încă un set de comenzi "standard" (chiar dacă e pe cale să "se nască" unul - ultimele produse încep să se alinieze la interfaţa CAS). Astfel încît fiecare cartelă fax trebuie să fie adresată cu un soft specific. De aceea, înainte de a cumpăra o cartelă fax, se recomandă să se facă o informare-documentare cît mai detaliată relativ la posibilităţile pe care softul aferent le oferă.

Pe lîngă diverse conversii, cartelele fax mai oferă şi alte facilităţi, întîlnite altfel numai la telefaxurile foarte scumpe. Astfel de exemplu comunicările ce nu sînt foarte urgente pot fi timise seara, după ora 18, cînd tarifele telefonice sînt mai convenabile - chiar dacă ele sînt pregătite dimineaţa, în timpul funcţionării normale a birourilor.

Tot atît de simplă este transmisia unor circulare: textul este "predat" cartelei fax o singură dată; aceasta îl expediază apoi tuturor destinatarilor. Selecţia fiecărui abonat - necesară la telecopiatoarele uzuale - este superfluă, fiind preluat de soft.

Un avantaj clar al PC-faxurilor este calitatea mai bună a transmisiei. Căci dacă la un echipament fax obişnuit, în mod necesar, prin scannare, apare o pierdere de calitate, cartela fax converteşte datele în format fax în mod optim. Dispare şi necesitatea imprimării prealabile a documentului - altă etapă care are ca efect o diminuare a calităţii imaginii de transmis. Astfel, pierderile de informaţie prin teletransmisie sînt reduse la un minim.

Şi faptul că folosind cartelele fax se poate lucra fără hîrtie - sau, cel puţin, fără hîrtia specială, scumpă şi poluantă, cu care lucrează telecopiatoarele obişnuite - constituie un avantaj important al cartelelor fax.

Suplimentar, multe cartele fax oferă prin softul aferent o carte de telefon a tuturor abonaţilor la serviciile fax. Astfel că se evită şi căutarea şi formarea manuală - uneori, incomodă - a numărului de telefon a destinatarului.

Adevărata lor putere, cartelele fax şi-o demonstrează însă de-abia într-o reţea, în care există un fax-server. În reţea, fiecare calculator conectat poate emite sau recepţiona comunicări prin fax. Dispecerizarea în timp a transmisiilor individuale este preluată de soft. Suplimentar, în felul acesta, o comunicare importantă este primită într-adevăr imediat după recepţia ei. Toate telefaxurile sosite sînt stocate pe server şi nu mai încarcă spaţiul disc existent pe calculatorul post de lucru. Ele pot fi arhivate centralizat, permiţînd o regăsire ulterioară lesnicioasă.

Unde-i multă lumină, mai apar şi umbre. În cazul cartelelor fax, "umbra" face ca telecopiatoarele "obişnuite" să nu fie "scoase" de pe piaţă sau în mod serios concurate: cartelele fax rămîn mai degrabă o completare. Căci, în rutina zilnică a oricărui birou apar o multitudine de documente care vor trebui să fie transmise ca faxuri obişnuite şi pe viitor - să amintim numai banalele semnături. Pentru a putea fi transmise prin intermediul PC-ului, aceste documente ar trebui scannate în prealabil. În acest caz, un telecopiator normal este, evident, mai comod. Mai ales că nici combinaţia cartelă fax + scanner nu este o alternativă pentru telecopiatoarele oferite la preţuri din ce în ce mai mici. Mai interesante - şi din ce în ce mai răspîndite - sînt soluţiile care

folosesc combinaţia cartelă fax + telecopiator obişnuit, acesta din urmă preluînd şi funcţia de scanner pentru PC, cînd este nevoie.

Cartelele fax mai au şi alte slăbiciuni. Cine vrea să poată fi găsit la orice oră, trebuie să aibă PC-ul pornit şi pregătit 24 de ore din 24. În plus, desigur, softul pentru controlul cartelei fax ocupă loc în memoria principală. Mulţi utilizatori vor fi semnificativ incomodaţi de lipsa celor 100-200 kocteţi pe care îi revendică, tipic, softul aferent unei cartele fax.

Cartela fax poate fi considerată ca alternativă pentru un telecopiator obişnuit numai în anumite cazuri, bine determinate şi destul de rare. Ca o suplimentare judicioasă însă, ea şi-a cîştigat de pe acum un loc în PC - nu înlocuind telecopiatorul "standard", ci oferind o sumă de servicii şi comodităţi suplimentare în avantajul utilizatorului de PC.

Listă de verificare

Dacă vă tentează achiziţionarea unei cartele fax, poate că e util să treceţi în revistă, punct cu punct, relaţia între ceea ce se oferă şi necesarul dumneavoastră, relativ la:

- autorizarea de funcţionare din partea poştei
- suprafaţa utilizator
- viteza de transmisie
- posibilităţile de conversie (PC Fax)
- imprimantele cu care poate lucra (tip, set de caractere, fonturi)
- posibilitatea de lucru în reţea (sisteme de operare, număr utilizatori)
- carte de telefoane şi adrese
- repetarea automată a apelului
- mixajul de texte, grafice şi imagini
- utilitare de scannare
- posibilităţile de editare (text, grafică)
- posibilităţile de înglobare în/ interfaţare cu aplicaţii deja existente
- inteligenţă proprie (procesor, RAM, etc.)
- transferul de fişiere
- posibilităţile de a lucra cu mai multe linii telefonice deodată
- necesarul de memorie principală
- raportul preţ/performanţă
- posibilităţile de utilizare a unui telecopiator autonom ca scanner şi/sau imprimantă
- posibilităţile de emisie la anumite ore, programate
- posibilităţile de emisie a circularelor
- funcţionarea în background
- jurnal de emisie/recepţie

ing. Fettich Iosif

Ce facilităţi trebuie să ofere softul aferent unei cartele fax?

Cu cît e mai simplă suprafaţa utilizator, cu atît mai bine. Transmisia prin telecopiere fiind în genere foarte simplă - introdu hîrtia, apasă butonul, formează numărul, gata - nici soluţia PC nu are voie să fie complicată - altfel, utilizatorul obişnuit va prefera în continuare varianta "clasică", cu utilizarea unui telefax obişnuit.

Un punct cîştigat în ce priveşte acceptarea de către utilizatorii finali îl reprezintă posibilitatea de a putea transmite (sau recepţiona) telefaxuri direct din cadrul aplicaţiei cu care lucrează de obicei - ceea ce implică posibilitatea de a include în aplicaţii module care să lucreze cu cartela fax. Tendinţa accentuată spre suprafeţe utilizator grafice face ca posibilitatea de a utiliza softul de fax de sub Windows să fie recomandabilă.

La transmisia de texte, ar trebui să existe nu numai varianta ASCII, ci şi emulări ale unor imprimante matriceale sau laser (Epson, HP Laserjet). Numai astfel se atinge calitatea textelor obţinute cu procesoarele de texte uxuale, făcînd posibilă utilizarea cartelelor fax pentru corespondenţa de afaceri.

Ca formate grafice, ar trebui să fie disponibile măcar formatele PCX şi TIFF. Posibilităţi de conversie în/din programe de grafică precum GEM, Dr. Halo, PC Paintbrush nu sînt neapărat necesare, dar uşurează editarea faxurilor. În acest context, este recomandabil şi suportul soft pentru lucrul cu mouse.

Pentru realizarea unor scrisori de afaceri care să corespundă calitativ, posibilitatea de-a amesteca fără restricţii grafice şi texte este o necesitate. În plus, trebuie să existe posibilitatea combinării mai multor documente (prospecte, oferte etc.) într-unul singur.

Posibilitatea stabilirii orei de transmisie - pentru a economisi din taxe - este o necesitate, la fel cum nu ar trebui să lipsească o carte de telefon. Dacă aceasta în plus poate fi împărţită în grupe, devine mai simplă transmisia unor circulare la anumiţi destinatari. Nu are voie să lipsească posibilitatea de a repeta în mod automat apelul, dacă acesta nu a reuşit din prima încercare.

Un jurnal detaliat al transmisiilor permite o administrare raţională a documentelor.

Mai ales pentru lucrul în reţea, funcţionarea în background a softului este indispensabilă.

Iar pentru ca aplicaţiile uzuale să poată fi folosite normal, necesarul de memorie rezidentă nu ar trebui să depăşească 100 ko.

Şi, nu în ultimul rînd, manualul de utilizare ar trebui să fie conceput didactic, constituindu-se într-o documentaţie uşor inteligibilă a funcţiilor oferite.

	FSE Discovery 2448HT	OAFax V3	FSE AMS ZX 1896	ARC Dynalink	JT Fax 4800	P+W Speedyfax
Preţ (DM)	318	575	598	690	900	1497
Cerinţe sistem						
Memoria principală (koct)	640	640	640	640	640	256
Slot	8 biţi	8 biţi	8 biţi	8	8 biţi	8 biţi
Versiune MS-DOS	3.0	3.3	3.0	3.0		
Date tehnice						
Utilizabil ca modem	da	da	da	da	da	
Cablu telefonic	2,1 m / RJ11	2m / RJ11	2m /RJ11	2m / RJ11		3 m / TAE-8
Funcţii soft						
Aplicaţie DOS		da			da	
Aplicaţie Windows		nu			nu	
Memorie ocupată ca TSR		72 ko			200 ko	
Preimprimate	nu	da	nu	nu		da
Transmisie de ecrane	nu	nu	nu	nu		da
Snapshot	da	da	da	da	da	nu
Intercepţia imprimării	da	da	da	da	da	da
Preconversie	da	da	da	da		da
Conversie automată	da	da	da	da		nu
Conversie în background	nu	nu	nu	nu		nu
Format -> faxformat	PCX, TIF, BFX, DCX	ASCII, PCX, GEM, TIF	PCX, TIF, BFX, DCX	PCX, DCX	ASCII, PCX	ASCII, PCX, TIF, IMG
faxformat -> Format			PCX	PCX	PCX	PCX, TIF, IMG
Viteza de transmisie	2400, 4800 bps	2400,4800, 7200,9600 bps	2400, 4800, 7200, 9600 bps	2400, 4800, 7200, 9600 bps	2400, 4800 bps	2400, 4800, 9600 bps
Moduri de transmisie	normal	normal/fin	normal	normal		normal / fin
Nivele de gri	nu	da	nu	nu		nu
Comprimare pagină	da (deselectabil)	da (deselectabil)	da (deselectabil)	da (deselectabil)		da (deselectabil)
Antet	da	da	da	da	da	da
Pagină-copertă	da (deselectabil)	nu	da (deselectabil)	da (deselectabil)		da (deselectabil)
Recunoaştere fax	nu	da	nu	nu		nu
Program de grafică	nu	da	nu	nu	da	da
Editor de texte		da			nu	da
Carte de telefon integrată		da	da		da	da
Transm. o-dată în timp	da	da	da	da	da	da
Circulare	da	da	da	da	da	da
Jurnal	Em	Em/Rec	Em/Rec	Em/Rec		Em/Rec
Polling	nu	da	nu	nu	nu	nu
Recepţie în background	da	da	da	da		da
Pr. princ. în background	da	da	da	da		da
Mouse	nu	nu	nu	nu		nu
Suport hard-scanner	nu	Microtek, Umax OADC	nu	nu	da	Canon I-12, HP, Logitech, Panaso
Suport hard - imprimante	HP-Laserjet, ESC-P compatibile	HP-Laserjet II Brother M1724, NEC 220, Ink-Jet BJ80 Epson 8/9/24 Pin	Hp-Laserjet, ESC/P compatibile	HP-Laserjet, ESC/P compatibile	ESC-P compatibile	HP-Laserjet, ESC-P compatibile
Informaţii suplimentare		A&M Computertechnik GmbH, Ötztaler Str. 18, 8000 München 70 tel. (089)743170, fx (089)74317100	FSE Computerhandels GmbH, Schmiedstr. 11, 6750 Kaiserslautern tel. (0631)67096 fx(0631)60697	ARC Computer-vertrieb GmbH, Freisinger Str. 3, 8057 Eching, tel. (08165)62899 fx(08165)64188	Interquad, 6078 Neulsenburg	P+W Electronic GmbH, Lortzingstr. 52, 7967 Bad Waldsee, tel. (07524)4508, fx(07524)4419

	Ferrari Fax	Dr. Neuhaus Faxy Master	ARC PCFax 9600 II	AEG Twinfax	PC Fax 2000	DNS Gammfax CP
Preţ (DM)	1704	1708	1980	3021	3400	3751
Cerinţe sistem						
Memoria principală (koct)	640	450		384		512
Slot	8/16 biţi	8 biţi	8	8 biţi		8 biţi
Versiune MS-DOS	3.1			3.0		3.3
Date tehnice						
Utilizabil ca modem	da	da	da	da		
Cablu telefonic	3m / TAE-8	5,15m / TAE-8		6,2m / TAE-8		2,15m / RJ11
Funcţii soft						
Aplicaţie DOS	limitat	da		da		
Aplicaţie Windows	da	nu		nu		
Memorie ocupată ca TSR	29 ko	103 ko		60	40 / 80 ko	
Preimprimate	da	da	da	da		nu
Transmisie de ecrane	nu	nu	nu	da		nu
Snapshot	nu	nu	nu	da	da	nu
Intercepţia imprimării	da	da	nu	da	da	da
Preconversie	nu	da	da	da		da
Conversie automată	nu	da (opţional)	nu	da		da (ASCII)
Conversie în background	nu	da	nu	nu		da (ASCII)
Format -> faxformat	ASCII, PCX, DCX	ASCII, CUT, PCX, IMG	ASCII, PCX, IMG, MSP	ASCII, PRN	PCX	ASCII, PCX, TIF, IMG,RST
faxformat -> Format	PCX, DCX	CUT, IMG, PCX	MSP, PCX	TIF, PCX, PCL, IMG	PCX	ASCII, PCX, TIF, IMG,RST
Viteza de transmisie	2400, 4800, 7200, 9600 bps	2400, 4800, 7200, 9600,	2400, 4800 7200, 9600 bps	2400, 4800, 7200, 9600 bps	2400, 4800, 7200, 9600 bps	2400, 4800 bps
Moduri de transmisie	normal / fin	normal/fin	normal	normal/fin/mic		normal
Nivele de gri	nu	nu	nu	nu		nu
Comprimare pagină	da (deselectabil)	da (deselectabil)	da (deselctabil)	da (deselectabil)		da (deselectabil)
Antet	da	da	da	da		da
Pagină-copertă	da (deselectabil)	nu	da (deselectabil)	da (deselectabil)		nu
Recunoaştere fax	nu	nu	nu	nu		nu
Program de grafică	nu	nu	da	da		nu
Editor de texte	da	da		da		
Carte de telefon integrată	da	da	da	da		
Transm. c-dată în timp	da	da	da	da	da	da
Circulare	da	da	da	da	da	da
Jurnal	Em/Rec	Em/Rec	Em/Rec	Em/Rec		Em/Rec
Polling	da	nu	da	da	nu	da
Recepţie în background	nu	da	da	da	da	da
Pr. princ. în background	nu	nu	da	da	da	da
Mouse	opţional	nu	opţional	opţional		nu
Suport hard-scanner	via Windows 3.0	Faxy SP, HP, Microtek, Panasonic	Canon, Chinon MS, Tele-Scan	telefax autonom	da	Canon IX-12, HP
Suport hard - imprimante	via Windows 3.0	Faxy SP, Fuji, Olivetti, Laserstar, HP-Laserjet IIP+III, Epson FX, LQ, Juki	HP-Laserjet II, IBM Grahics Pr., Canon BJ-80,130, Epson 9/24 pin Brother M-1724	ESC/P compatibil, IBM Graphics Printer	Epson	HP-Laserjet+, IBM Graphics Pr., IBM Proprinter, Xerox 4045, Canon, Epson,NEC

Precizări relative la tabelul comparativ

Preţ - preţurile sînt preţuri de vînzare recomandate; cu reduceri, este posibil să puteţi obţine echipamentele şi mai ieftin. În preţ este inclusă obligatoria taxă de valoare adăugată, fixată prin lege (în Germania) la 14%. (Cumpărătorilor ce duc peste graniţă bunurile cumpărate, li se returnează contravaloarea acestei taxe, la prezentarea dovezii corespunzătoare de la vamă).

Memoria principală - dimensiunea minimă a memoriei de bază necesară softului anexă pentru o funcţionare ireproşabilă. Datele sînt cele declarate de producători.

Slot - se referă la tipul slot-ului liber în care urmează să se plaseze cartela fax la instalarea în PC

Versiune MS-DOS - cea mai mică versiune DOS cu care softul aferent cartelei DOS poate lucra. Mai ales laptop-urile şi notebook-PC-urile se livrează cu versiuni MS-DOS foarte vechi (dar foarte economicoase în ce priveşte consumul de memorie). Atenţie, marea majoritate a programelor fax necesită versiuni >= 3.0.

Utilizabil ca modem - uneori, cartela fax este de fapt o cartelă fax-modem, putînd fi folosită şi ca modem (de regulă, cu restricţii, mai mult sau mai puţin importante).

Cablu telefonic - dimensiunea cablului anexat la livrare, împreună cu tipul ştecherului (TAE-8 - standard german, RT11 - standard american; la noi, încă nu există nici un fel de standardizare în această direcţie; la instalare se fixează legăturile necesare cu şuruburi ...)

Aplicaţie DOS/Windows - precizează dacă softul aferent cartelei funcţionează ca aplicaţie DOS, sub Windows, sau şi-şi.

Memorie ocupată ca TSR - în cazurile cînd programul poate lucra ca program rezident, în background, specifică memoria "furată" de către acesta.

Preimprimate - o funcţie utilă a unor programe fax este inserarea opţională a unor antete de scrisoare sau logo-uri de firmă, astfel încît faxul ajuns la destinatar să arate ca o scrisoare "obişnuită".

Transmisie de ecrane (screencapture)- în softurile fax rezidente, există uneori posibilitatea ca o simplă combinaţie de taste să aibă ca efect transmisia ca fax a ecranului curent.

Snapshot - este vorba de posibilitatea de a stoca, într-un fişier, ecranul curent, pentru eventuale prelucrări ulterioare.

Intercepţia imprimării (printcapture) - cu această funcţie, orice program de prelucrare texte poate fi folosit pentru pregătirea de faxuri. Un mic utilitar rezident deturnează datele trimise spre imprimantă, orientîndu-le către cartela fax.

Preconversie - este vorba de conversia prealabilă a fişierelor ce se vor transmite, înainte de stabilirea legăturii - dacă un program nu are această funcţie, va trebui să acceptaţi eventuali timpi de aşteptare inutili, pe timpul transmisiei.

Conversie automată - la unele programe, înainte de transmisie, datele trebuiesc convertite "manual". Este desigur mai comod dacă această conversie se face automat, fără intervenţia utilizatorului.

Conversie în background - unele programe convertesc fişiere ASCII în mod automat înaintea transmisiei, fără a bloca calculatorul pe durata acestei conversii.

Format - > faxformat - precizează care sînt formatele grafice recunoscute prin soft.Dacă vreţi să transmiteţi şi grafică, nu numai texte, atunci softul aferent cartelei fax trebuie să poată prelucra fără probleme formatul grafic în care i se prezintă datele. (Altfel, este necesară imprimarea prealabilă şi scannarea imaginii ce urmeazĂ a fi transmisă). Softuri performante de fax ar trebui să "ştie" cît mai multe formate grafice.

faxformat -> Format - precizează care sînt formatele grafice în care poate fi convertit un fax recepţionat

Viteza de transmisie - 9600 bps sînt aproape standard în domeniul faxurilor actuale. Pentru legături telefonice foarte proaste, ar trebui însă să existe şi posibilitatea trecerii la viteze de transmisie mai mici. De reţinut diferenţa între bps (biţi pe secundă) şi Baud. Ultima unitate de măsură determină numai numărul de schimbări de frecvenţă (pe secundă) pe linia telefonică. Începînd cu 600bps, la o singură schimbare de frecvenţă se transmit mai mulţi biţi.

Moduri de transmisie - cu cît e mai mare rezoluţia unei pagini transmise prin fax, cu atît calitatea ei este mai bună. Dar rezoluţii mărite necesită şi timpi de transmisie corespunzător mai mari. Pentru texte obişnuite, rezoluţia normală e suficientă.

Nivele de gri - cartelele fax cele mai noi prelucrează graficele nu numai ca şi combinaţii de puncte albe şi negre, ci pot transmite şi diferite nuanţe de gri.

Comprimare pagină - cartelele fax din grupa III oferă un procedeu de compresie care permite reducerea timpilor de transmisie.

Antet - începînd cu norma fax pentru grupa III, este obligatoriu pe documentul transmis un rînd în care se specifică data, numărul de apel şi numele expeditorului.

Pagină - copertă - la transmisia mai multor pagini, prima pagină ar trebui să conţină date relative la numărul total de pagini precum şi date exacte privind expeditorul şi destinatarul. Desigur, este avantajos ca această funcţie să fie integrată în programul fax aferent. Totodată, la cerere, ea ar trebui să poată fi invalidată.

Recunoaştere fax - unele cartele fax "ridică receptorul" la sosirea unui apel telefonic, aşteptînd un semnal de la chemător. Dacă acest semnal nu vine într-un interval scurt, echipamentul anunţă apelul

utilizatorului, eliberînd linia telefonică pentru o convorbire normală.

Program de grafică (integrat) - cu aceste editoare grafice minimale integrate, utilizatorului i se dă posibilitatea, de ex., să adnoteze paginile fax recepţionate cu scurte notiţe.

Editor de texte (integrat) - permite editarea minimală a textului ce urmează a fi transmis, direct, fără intermediul unui procesor de texte "standard".

Carte de telefon integrată - multe programe fax "vin" împreună cu cartea de telefon a abonaţilor serviciului telefax, permiţînd astfel găsirea rapidă şi comodă a numărului dorit.

Transmisie comandată în timp - poate fi avantajos uneori ca transmisia unui document să nu se facă imediat, ci la o anumită oră (sau dată) - pentru a beneficia de exemplu, de tarifele reduse din timpul nopţii.

Circulare - cînd un fax trebuie să ajungă la mai mulţi participanţi, în programele fax bune este suficientă elaborarea listei destinatarilor - restul acţiunilor le rezolvă softul singur, fără intervenţia utilizatorului.

Jurnal - mai ales la cartelele ce se utilizează în domeniul comercial, jurnalul (evidenţa) protocolului de utilizare (emisii-recepţii) este un auxiliar indispensabil pentru facturare.

Polling - această funcţie, utilizată rareori, permite preluarea unor faxuri disponibile de la un alt abonat. Echivalent (oarecum) al serviciilor gen "post-restant" sau "cutie poştală".

Recepţie în background - cele mai multe programe fax lucrează şi în background, dar sînt puţine cele ce permit continuarea lucrului cu calculatorul şi pe durata recepţiei (cel mai adesea, utilizatorul e pus în aşteptare - nu i se solicită însă nici o intervenţie, recepţia se desfăşoară "pe nevăzute").

Programul principal lucrează şi în background - dacă cartela fax trebuie să fie gata de recepţie 24 de ore din 24, atunci programul fax trebuie să poată lucra rezident, în background.

Mouse - precizează dacă programul fax poate sau nu să fie utilizat folosind un mouse.

Suport hard/scanner - pentru transmisia de pagini scrise de mînă sau a unor desene, este necesară scannarea anterioară a acestora, folosind un scanner. Uneori, ca scanner poate fi utilizat chiar un telecopiator "adevărat" (autonom).

Suport hard/imprimantă - bineînţeles, trebuie să puteţi "scoate la imprimantă" paginile recepţionate prin fax. Dacă documentul nu este sec, ci are şi informaţii de formatare (sublinieri, îngroşări, diferite tipuri de caractere, etc.) atunci acestea trebuiesc transmise imprimantei în aşa fel încît să imprime textul într-o formă corectă - evident, cel care vă trimite faxul nu prea poate şti ce fel de imprimantă aveţi în sistem. Din păcate, datele furnizate de producători la acest punct sînt, uneori, foarte superficiale.

Grupele fax

Telecopiatoarele (telefaxurile - prin extensie, prin telefax sau fax se înţelege atît echipamentul care realizează transmisia, cît şi documentul transmis) se categorisesc în 4 grupe. Aceste grupe se deosebesc prin caracteristici de bază; pentru utilizatori, diferenţele "se văd" în viteza şi calitatea transmisiei.

Telecopiatoare din grupa I

Această grupă este caracterizată de o tehnică depăşită de transmisie şi este neinteresantă pentru noi - în ţară, se pare, nu sînt instalate astfel de telefaxuri.

Telecopiatoare din grupa II

Şi în acest caz este vorba de o tehnică foarte veche. Informaţiile sînt prelucrate analogic, ca şi în cazul grupei I, ceea ce face ca echipamente fax din grupa II să mai existe foarte puţine. Viteza lor de lucru redusă (cca. 3 minute pentru o pagină A4) precum şi insuficienta corecţie a erorilor le fac neinteresante pentru aplicaţiile actuale.

Telecopiatoare din grupa III

sînt cele mai răspîndite în prezent şi lucrează cu tehnici numerice. Deoarece se face o compresie a datelor, timpul de transmisie variază funcţie de conţinutul documentului. Compresia datelor se face fie unidimensional (orizontal, pentru fiecare rînd scannat) fie bidimensional (orizontal şi vertical) şi are rolul de a minimiza timpul necesar transmisiei. Astfel, o pagină A4 cu un conţinut obişnuit poate fi transmisă pe linia telefonică în circa 1 minut. Viteza fizică pentru faxurile din grupa III este stabilită de regulă de normele CCITT V.27 - deci 2400 şi 4800 biţi pe secundă. Telecopiatoarele mai noi se conformează şi normelor CCITT V.29, pentru 7200 şi 9600 bps. Rezoluţia normală pentru echipamentele din această grupă este de 203 x 98 dpi (dots per inch - puncte pe inch). Deseori, rezoluţia pe verticală poate fi mărită pînă la 196 dpi, iar pe alocuri există deja echipamente ce lucrează cu 400 x 400 dpi.

Telecopiatoare din grupa IV

Aceste echipamente sînt construite pentru legături digitale, lipsite de erori de transmisie, cum sînt asigurate de exemplu în ISDN. Datele de transmis sînt comprimate la maxim şi transmise apoi receptorului. Echipamentele din această categorie vor deveni interesante de-abia cînd ISDN va fi disponibil pe scară largă, în toată lumea - lucru improbabil în viitorul apropiat.

Fonturi

Luaţi un borcan de marmeladă, de pe un raft cu bunuri de consum, dintr-un magazin ecologic occidental, şi exeminaţi-i eticheta: un scris de copil uşor nesigur - ca de pe tabla unei şcoli. Eticheta n-a fost în nici un caz scrisă de mînă, ci a fost tipărită. De ce s-a folosit atunci atunci acest scris de tablă de şcoală? Este foarte simplu: acest caracter de literă semnalizează: "Pur natural - în întreprinderea noastră totul este lucrat de mînă, deci şi scris de mînă". Deoarece cel mai adesea produsele nu pot fi gustate înainte de a fi cumpărate, inscripţionarea de pe ambalaj trebuie nu numai să-i lase clientului "gura apă" ci şi să-i cîştige încrederea.

Acelaşi lucru se întîmplă şi în prospectul unui aparat CD (Compact Disk) produs de o anumită întreprindere High - Tech - caracterul de literă (fontul) utilizat în acest caz: Helvetica, fără serife (deci fără linii orizontale sau cîrlige la sfîrşitul caracterelor). Eleganţa simplă, lipsită de podoabe semnalizează precizia cultivată şi competenţa tehnică.

Două exemple din care se poate observa că se foloseşte un caracter de literă pentru a transmite un mesaj; caracteristicile estetice ale unui caracter de literă nu reprezintă neapărat un scop artistic în sine, ci sînt folosite în anumite scopuri. Corespunzător cu aceasta există un număr aproape de necuprins de caractere de literă şi de familii de caractere de literă diferite. Toate, însă, nu doresc decît un singur lucru: să-l ia pe cititor de mînă şi să-l conducă prin text (sau să-l rătăcească, în funcţie de morală).

Doriţi să realizaţi o invitaţie elegantă, o listă de bucate împopoţonată sau o broşură cu calculatorul? pentru ca ea să arate într-adevăr bine, aveţi nevoie de o colecţie corespunzătoare de familii de litere şi de diferite dimensiuni de literă. Procesorul D-vs. de texte vă oferă suficiente alternative? Atunci desigur că n-aţi avut nefericita experienţă, ca de exemplu textul unui antet să nu fie tipărit cu acelaşi caracter de literă ca şi textul cursiv - chiar dacă imprimanta folosită ar fi putut să facă acest lucru.

Astfel, de exemplu, procesorul de texte "Word 5.0", chiar lucrînd cu imprimanta laser HP Laserjet, nu cunoaşte pentru caracterul de literă Courier decît dimensiunile 10 şi 12 puncte (punct: înălţimea unui caracter. 1 punct = 0,335 mm). Dacă pentru text se utilizează dimensiunea 12 puncte, şi se doreşte o scriere mai marcantă lipsesc dimensiunile de scriere mai mari (de exemplu de 18 sau 24 de puncte).

Problemele nu apar numai la titluri: dacă doriţi să imprimaţi o notă de subsol, cu caracterul de literă Courier, atunci dimensiunea de 10 puncte nu este suficient de mică.

Dar nu trebuie să renunţaţi numaidecît, dacă trebuie să scrieţi un referat sau o disertaţie cu Word: există mijloace şi căi de a utiliza şi alte caractere de literă şi dimensiuni, şi aceasta nu numai pentru Word, ci pentru majoritatea aplicaţiilor. Cum puteţi ajunge la noi caractere de literă pentru calculatorul D-vs., cît costă şi cum le puteţi utiliza optimal, veţi putea afla din paginile care urmează. În plus vor fi reprezentate fundamentele temei familii de caractere de literă şi multe sfaturi folositoare.

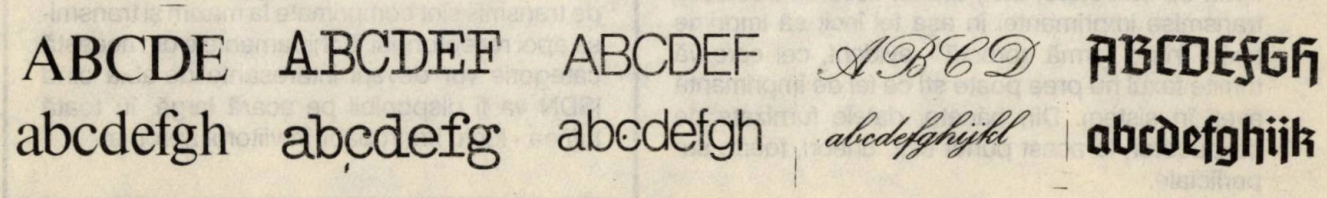
Utilizarea corectă a scrierilor

Cine doreşte să utilizeze efectul optic al diferitelor familii de caractere, trebuie să ştie să le folosească. Utilizarea unei provizii bogate de familii de caractere, aşa cum oferă calculatoarele cu ajutorul programelor DTP (DTP = Desk Top Publishing - programe pentru tehnoredactare), în lipsa unei know-how corespunzător, conduce cel mai adesea doar la rezultate de diletanţi.

Una din regulile de bază spune: "Să nu utilizezi niciodată prea multe caractere de literă pe aceiaşi pagină". Hotărîtoare este alegerea unui caracter de literă şi-abia apoi - variaţiile ei (mare/mică, îngroşată, cursiv, etc.). Problema este să se constituie mai întîi o colecţie cît mai mare de caractere de literă din care să se poată opera selecţia. Dar să vedem mai întîi ce probleme apar la prelucrarea scrierilor pe calculator.

Scrierile şi calculatorul

Pentru ca utilizatorul unui calculator să poată utiliza un anumit tip de caracter de literă, procesorul de texte respectiv trebuie să cunoască scrierea respectivă. Mai



Times | Text de maşină de scris | Helvetica | Scris de mînă | Fraktur

întîi trebuie pregătit, aşadar, caracterul de literă dorit pentru a putea fi utilizat din procesorul de texte dorit şi pentru a putea fi tipărit la imprimantă.

Pentru a putea reprezenta pe ecran un caracter, sau pentru a-l putea trimite la imprimantă, din interiorul unui procesor de texte, caracterele trebuie transformate după o schemă "înţeleasă" de calculator. Un caracter devine inteligibil pentru un PC, respectiv pentru un soft, de îndată ce a fost transformat într-un mănunchi de date, deci după ce a fost codificat în biţi şi bytes. În acest mănunchi de date este cuprinsă alcătuirea caracterului dintr-o anumită ordonare a unor puncte individuale. Aceste puncte sînt plasate într-un domeniu bine definit (de exemplu un bloc de 28 de puncte pe orizontală şi 36 de puncte pe verticală), acesta constituind aşa-numita "matrice caracter".

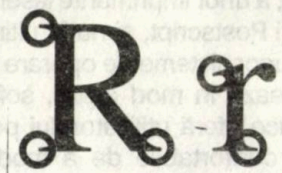
Pentru fiecare caracter tastat, la tastatura PC-ului, programul determină calculatorul să reprezinte respectivul caracter într-o anumită poziţie de pe ecran într-o astfel de matrice-bloc. Mai multe astfel de matrice-bloc una după alta compun un şir de caractere, un cuvînt, sau şi mai mult un întreg text.

Un procesor de texte trebuie să poată accede la un set complet de caractere, pentru a putea construi pe baza acestor "metode" caracterul tastat, pentru a crea în acest mod un text şi al stoca într-o formă reutilizabilă pe un harddisk sau pe o dischetă.

Care sînt scrierile utilizabile pe calculator?

Un caracter de literă care poate fi reprezentat foarte uşor într-o matrice pentru calculator este Courier, creat de IBM. Acest caracter de literă a fost conceput în mod special pentru reprezentarea pe monitoare, la sfîrşitul anilor 50. Rezoluţia disponibilă în acel moment era de 320x200 puncte - corespundea deci standardului CGA. Din cauza acestor limitări, în cazul caracterului de literă Courier, s-a renunţat la multe caracteristici, cum ar fi grosimi diferite de linii şi rotunjimi la colţuri, care sînt tipice pentru multe alte familii de caractere.

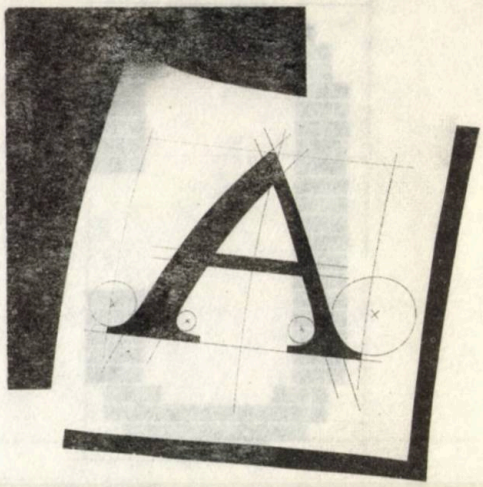
Majoritatea celorlalte scrieri au fost elaborate manual înainte de apariţia calculatoarelor, fără să ţină cont de limitările impuse de acestea. Cîteva dintre acestea sînt scrieri decorative şi ornamentate. Ele dispun - ca şi toate celelalte familii de caractere create de-a lungul timpului cu creionul şi pana - de o mulţime de caracteristici de prezentare. Ele dispun de rapoarte de mărime diferite în partea superioară, de mijloc şi de jos; multe caractere avînd şi anumite serife (linii orizontale sau cîrlige la părţile terminale ale literei). La acestea se adaugă diferite grosimi de scriere în interiorul unui caracter. Toate acestea îngreunează transcrierea în biţi şi bytes - şi doar aşa pot fi prelucrate de calculator.



Serife

Scanarea unei scrieri nu este practicabilă

Cu toate că o scriere poate fi "citită" cu ajutorul unui scanner, codificată sub forma unui rastru de puncte, transformată în biţi şi reprezentată pe monitor, de exemplu cu ajutorul unui program de pictură (cu ajutorul cărora s-ar putea elabora chiar caractere de literă proprii), aceste date nu pot fi prelucrate în continuare ca nişte caractere obişnuite de către un procesor de texte. Pentru aceasta



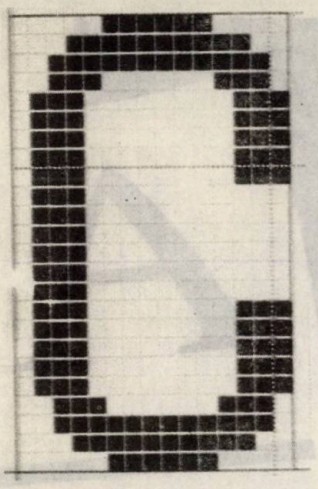
ar trebui soft-uri suplimentare - calea este prea puţin practicată.

Programele de recunoaştere optică a caracterelor create pentru scanner-e pot citi caractere, atît timp cînd sînt elaborate relativ simplu (altfel pot apare erori de identificare), programele moderne putînd fi chiar "învăţate" să recunoască şi scrieri decorative. Dar scrierile recunoscute nu sînt redate în aceeaşi formă în care au fost citite, ci este utilizată o formă standard pentru reprezentarea lor pe monitor sau pentru tipărirea lor la imprimantă. Această soluţie poate fi utilizată la introducerea unor volume mari de texte, dar nu şi pentru îmbogăţirea repertoriului de scrieri.

Două căi pentru producerea de scrieri utilizabile pe calculator:

Fonturi - bitmap

Între timp s-au stabilit două tehnici de bază. Una funcţionează în mod asemănător, ca în primul nostru exemplu, caracterul de literă Courier: fiecare caracter este construit dintr-o succesiune de puncte (pixeli) poziţionate pe un rastru predefinit. Caracterele sînt reprezentate ca "hărţi de biţi" (bitmaps) orientate pe pixeli - cea mai simplă metodă de a crea noi scrieri (numite şi fonturi) pentru calculator - de aici şi denumirea de fonturi bitmaps.



Font bitmap: construcţie din puncte

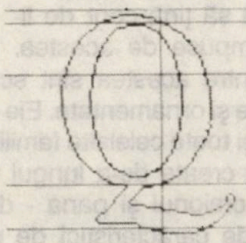
Această metodă este recomandabilă pentru caractere "pline" (caractere care nu sînt alcătuite doar din linii). Trebuie luată în considerare, însă, şi existenţa cîtorva dezavantaje, cum ar fi apariţia aşa-numitului efect treaptă, sau dinţi de fierăstrău, pe conturul unor astfel de caractere. Acest dezavantaj poate fi contracarat prin utilizarea unui rastru mai fin, dar fiecare pixel care intră în plus în componenţa unui caracter, necesită spaţiu suplimentar de memorie - ceea ce implică utilizarea unui hard mai pretenţios, fapt care duce la creşterea costurilor şi se loveşte de limitele tehnicii.

Fonturi outline

Efectul dinţilor de fierăstrău, obţinut la mărirea unui caracter bitmap, poate fi înlăturat prin utilizarea tehnicilor de prezentare vectorizată a caracterelor.

Această tehnică presupune un efort suplimentar de programare şi de calcul în comparaţie cu tehnica bitmap, şi presupune adesea şi existenţa unui hard mai puternic. În cazul caracterelor de literă vectoriale nu mai trebuie definite toate punctele imaginii, ci doar punctele de început şi de sfîrşit ale unei linii drepte. Pentru o formă tridimensională sînt necesare trei coordonate, pentru un patrulater patru, un cerc fiind aproximat printr-un număr cît mai mare de segmente de dreaptă. Cu cît sînt utilizate mai multe puncte pentru descrierea unei serife sau a unei rotunjiri, cu atît mai exact va putea fi reprezentat caracterul pe ecran sau pe hîrtie.

Un exemplu îl constituie aşa-numitele fonturi outline. Aceasta înseamnă că aceste caractere sînt descrise numai prin liniile exterioare care urmăresc contururile.



Font outline: Caracterele sînt descrise prin contur

Abia cu dezvoltarea unui hard puternic şi cu apariţia unor calculatoare corespunzătoare (Apple Macintosh sau calculatoare personale cu microprocesoare Intel 80386), a unor sisteme grafice puternice, a unor imprimante laser cu facilităţi Postscript, şi nu în ultimul rînd a unor sisteme de operare care lucrează în mod grafic, softuri puternice oferă utilizatorului posibilităţi confortabile de a modela scrieri.

Pentru ca imprimantele să poată tipări fonturile este necesar un software de comandă special: driverul de imprimantă (situaţie valabilă şi pentru fonturile standard).

Fonturi la pachet

Pentru a dispune de mai multe caractere de literă în programele utilizate se pot cumpăra pachete de fonturi. Astfel de pachete oferă, de exemplu, firmele Adobe şi Bitstream. Pachetele lor de fonturi au fost elaborate în mod special pentru a fi utilizate în programele DTP, cum ar fi: "Ventura Publisher", "Pagemaker" sau "Quark Express". Avantajul: utilizatorul poate încărca noile fonturi direct în programul dorit, fără alt efort de instalare. Dezavantaj: un astfel de pachet este limitat la cîteva caractere de literă şi dimensiuni caracter. Aceste produse presupun doar existenţa programelor DTP, şi nu necesită nici o suprafaţă utilizator specială - este suficient sistemul de operare MS -DOS.

Suprafeţe utilizator grafice

În învechitele moduri caracter (mod text, nu grafic) ale primelor generaţii de PC-uri, rezoluţia monitoarelor şi a imprimantelor nu era suficientă pentru a putea oferi utilizatorului posibilitatea de a modela fonturi.

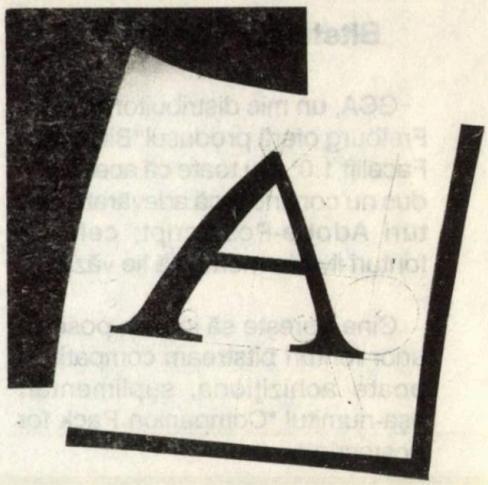
Datorită suprafeţelor utilizator grafice (Macintosh, pentru PC "Windows" sau "Geoworks Ensemble") utilizatorul vede caracterele pe ecran la fel cum vor arăta la imprimantă. Aceasta înseamnă că noile suprafeţe utilizator permit aşa - numita reprezentare WYSIWYG (What You Su Is What You Get): ceea ce se vede pe ecran corespunde cu ceea ce se va vedea la imprimantă. Abia aceste medii de programare oferă utilizatorului posibilitatea folosirii unui număr mai mare de fonturi şi nu numai ca fonturi bitmap ci şi ca fonturi vectoriale.

O soluţie convenabilă: programele Shareware

În multe cazuri, o soluţie convenabilă o oferă programele din sectoarele Public Domain (PD) şi Shareware. Paleta de funcţii a acestor programe se limitează, adesea, la funcţiile de bază. De exemplu există programe utilitare care permit utilizatorilor procesorului de texte Word, utilizarea alfabetului chirilic, care nu apare ca font în repertoriul programului, şi textele create pot fi listate chiar şi pe imprimante matriciale Epson.

Mai multe caractere de literă pentru textele D-voastră:

Generatoarele de fonturi



Cu ajutorul generatoarelor de fonturi, utilizatorul îşi poate crea caractere de literă şi dimensiuni de literă în limite largi.

Bitstream - Fontware

Bitstream este o firmă cu o tradiţie îndelungată în proiectarea şi realizarea de caractere de literă. "Bitstream Fontware" este un pachet combinat, alcătuit dintr-un pachet de scrieri şi un generator de fonturi. Programul de instalare furnizat copiază toate fişierele necesare pe harddisk. Din scrierile disponibile pe dischetă pot fi elaborate caracterele de literă la dimensiunile dorite specifice imprimantei şi modului grafic în care lucrează calculatorul. Generarea propriu-zisă, atunci cînd selecţia de scrieri şi de dimensiuni este mare, poate să dureze cîteva ore.

Programul permite o operare foarte uşoară. Pe monitor sînt afişate două ferestre de meniuri, în fereastra din stînga sînt afişate fonturile, iar în cea din dreapta dimensiunile de literă posibile.

Toţi paşii de lucru (selecţie şi generare) pot fi parcurşi prin intermediul opţiunilor situate sub cele două ferestre. Programul este livrat ca utilitar şi în furnitura programului DTP "Pagemaker 3.0"

Bitstream - Fontware
Ofertant: EDTZ,
8012 Ottobrunn bei München
Preţ: cca. 700 DM
Sistem de operare: MS-DOS

Typografica

Firma Schneider Data din Freising oferă generatorul de fonturi numit "Typografica". Typografica este vîndut ca produs individual, sau, în unele situaţii, împreună cu alte programe de aplicaţie (de exemplu "Timeworks Publisher").

După instalare, generatorul de fonturi este adaptat imprimantei şi modului grafic utilizat. Abia după definirea mediului de lucru hard şi soft, pot fi selectate, din ferestrele de lucru, fonturile dorite. Spre deosebire de Bitstream - Fontware, Typografica nu poate prelucra la un moment dat decît un singur caracter de literă. După selectarea acestuia urmează specificarea stilului de literă şi a dimensiunii. Informaţiile şi opţiunile suplimentare au fost astfel concepute încît aproape că se poate lucra fără manual de utilizare.

Typografica
Ofertant: Schneider Data
8050 Freising
Preţ: cca. 130 DM
Sistem de operare: MS-DOS

Fontware

Cu Windows 3.0 s-a scris un nou capitol din romanul foileton al suprafeţelor utilizator grafice pentru PC-urile MS-DOS. Windows oferă soluţia cvasi-ideală pentru astfel de programe "fontware".

Cu ajutorul programelor fontware, fiecărei aplicaţii Windows îi sînt puse la dispoziţie fonturi liber scalabile, care vor arăta la fel şi în reprezentarea de pe ecran şi la imprimantă.

Adobe Type Manager 1.0

"Adobe Type Manager" (ATM) a fost croit special pentru facilităţile suprafeţei utilizator grafice Windows 3.0. Furnitura de livrare conţine 13 fonturi diferite din familiile Courier, Helvetica, Times şi Symbol. Alte 22 de fonturi sînt oferite în pachetul suplimentar "Adobe Plus Pack". În urma instalării de pe cele 3 dischete cu cele 35 de fonturi şi soft-ul de comandă corespunzător (constînd din aşa-numitele Control Panel şi un driver Windows) se ocupă pe harddisk, 1,5 MByte.

În cursul procesului de instalare driverul este adăugat la fişierul "SYSTEM.INI". Lucrul cu ATM este foarte simplu: la nevoie se poate deschide un mic cîmp de control de sub Windows, pentru a şterge, de exemplu, fonturi. Altfel ATM lucrează neobservat în fundal.

Toate aplicaţiile Windows profită de ATM, şi mai ales aplicaţiile DTP cum sînt "Page Maker" sau "Ventura Publisher 3.0". Caracterele de literă sînt reprezentate corect pe ecran şi se caracterizează printr-o calitate ridicată la tipărire. Mai ales lucrul cu imprimante laser compatibile HP funcţionează excelent, rezultatul apropiindu-se de calitatea Postscript. Suplimentar fonturilor oferite de ATM pot fi încărcate toate fonturile Adobe-Type-1.

Fontware
Ofertant: Letraset, 6000 Frankfurt
Preţ: pachetul de bază cca. 280 DM, cu pachetul suplimentar optimal: 599 DM
Sistem de operare: MS-DOS

Bitstream Facelift 1.0

GCA, un mic distribuitor soft din Freiburg oferă produsul "Bitstream Facelift 1.0". Cu toate că acest produs nu conţine încă adevărate fonturi Adobe-Postscript, cele 13 fonturi livrate merită să fie văzute.

Cine doreşte să intre în posesia unor fonturi bitstream compatibile poate achiziţiona, suplimentar, aşa-numitul "Companion Pack for Postscript".

Programul este livrat pe 4 dischete şi după instalare ocupă 2,7 Mbyte pe harddisk. Cele două drivere ale programului sînt adăugate la "WIN. INI" şi "SYSTEM.INI" ca aşa-numite PS-Outline (PS semnifică Postscript) pentru 13 fonturi ne-Postscript.

De sub Windows, Facelift este controlat prin intermediul unei ferestre proprii. Cu ajutorul diferitelor icon-uri din fereastra activată se stabilesc fonturile care urmează a fi disponibile.

Dacă sub Windows sînt instalate mai multe imprimante, se poate stabili care dintre ele să lucreze cu Facelift şi care nu. Programul lucrează în fundal. O imprimare cu Facelift durează ceva mai mult decît în cazul concurentului său Adobe, în schimb calitatea fonturilor este deosebită.

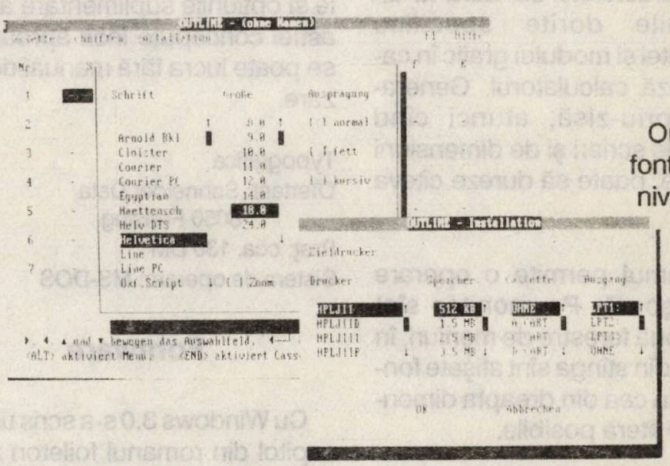
Bitstream Facelift
Ofertant: GCA, Freiburg
Preţ: cca. 300 DM (pachetul de bază), cca. 400 DM cu Companion Pack.
Sistem de operare: MS-DOS

Programe de fonturi la nivel DOS

Outline

"Outline" este un program care lucrează la nivel DOS. Programul oferă cîteva particularităţi: el este asemănător ca principiu cu un cartridges, deci cu o extensie hard pentru o imprimantă laser. Outline nu este totuşi decît un produs pur soft care utilizează conceptul fonturilor soft al imprimantelor laser compatibile HP, Kyocera, Brother sau Sharp. Extensiile pot fi adăugate pentru o serie întreagă de texte cunoscute (Wordperfect, Word, PC-Text, etc.).

La instalare, utilizatorul stabileşte fonturile şi dimensiunile care vor fi utilizate în diferitele procesoare de texte. Dimensiunile de literă nu mai pot fi modificate ulterior. Fonturile şi dimensiunile selectate ca imprimanta să poată interpreta corect instrucţiunile de reprezentare, programul generează simultan şi un driver de imprimantă care este încărcat automat înainte de imprimare.



Outline: fontware la nivel DOS

Outline oferă pentru diferitele fonturi o serie întreagă de efecte speciale: scrieri cu umbre, litere umbrite şi chiar scrieri 3D. Datorită faptului că se utilizează o metodă de compresie, fişierele de fonturi Outline necesită cu cca. 60% mai puţin spaţiu pe disc decît fonturile compatibile HP.

Fonturile sînt liber scalabile între 4 şi 144 de puncte şi stau la dispoziţie ad-hoc, în formatul dorit, pe imprimantele Laserjet III.

Outline
Ofertant: S.A.X., 7500 Karlsruhe
Preţ: cca. 1100 DM
Sistem de operare: MS-DOS.

Laser Star

"Laser Star" produs al firmei Star Division din Luneburg face parte de regulă din furnitura de livrare a imprimantelor din seria Brother HL-8 dar este oferit şi separat.

Programul poate încărca în imprimanta laser pînă la 50 fonturi soft diferite. Metoda este uluitor de simplă, în loc să genereze mai întîi fonturile soft, programul generează un driver imprimantă care recunoaşte cele 50 de fonturi. Utilizatorul trebuie să specifice mai întîi doar aplicaţia în care doreşte să utilizeze aceste fonturi, programul generînd driverul să utilizeze aceste fonturi, programul generînd driverul necesar şi încărcîndu-l în aplicaţia dorită.

Cele 50 de fonturi soft sînt copiate pe harddisk şi sînt încărcate la nevoie. Cu aşa-numita operare "download" aceste fonturi sînt încărcate cît se poate de rapid şi transmise imprimantei. Cu toate că Laser Jet se limitează astfel la un singur program de aplicaţie, pentru acesta sînt disponibile toate variaţiile posibile cu cele 50 de fonturi.

Laser Star
Ofertant: Star Division, 2120 Lüneburg
Preţ: cca. 500 DM
Sistem de operare: MS-DOS

Mai multe caractere de literă pentru textele D-voastră:

Emulatoarele Postscript

Cine nu doreşte să-şi cumpere o imprimantă Postscript scumpă, îşi poate comanda imprimanta cu ajutorul unor soft-uri speciale.

Postscript este un limbaj de programare elaborat pentru comanda aparatelor electronice de ieşire (v. "if" 3/90). Acest limbaj de programare a fost botezat Postscript la înfiinţarea firmei Adobe System Inc. în 1982 şi este o dezvoltare a limbajului "Impress".

Cu ajutorul unui interpretor Postscript, textele şi graficele pot fi imprimate în orice formă imaginabilă. Modul de lucru: Postscript 'descrie" imprimantei modul în care trebuie să arate o pagină întreagă şi nu doar o linie de text. O pagină prelucrată cu Postscript poate conţine o multitudine de elemente: mai întîi trebuie să fie specificat un sistem de coordonate, în raport cu care sînt posibile tot felul de rotiri, modificări de proporţii (translaţii) şi scalări, ale diferitelor fonturi. Toate caracterele obţinute vor fi tratate de Postscript ca şi grafice individuale. Ulterior, cu ajutorul acestui interpretor, toate obiectele pot fi reproduse la o rezoluţie corespunzătoare, aceasta putînd depăşi 2400 dpi (dot per inch). Aceste rezoluţii ridicate nu pot fi obţinute decît pe exponatoare speciale, cum ar fi de exemplu Linotype. Imprimantele laser moderne ating astăzi o rezoluţie maximală de 600 pînă la 300 dpi. Şi totuşi aceste imprimante pot utiliza Postscript. Limbajul poate fi implementat direct în hard-ul imprimantei, caz în care imprimanta converteşte comenzile primite de la calculator, înainte de imprimare.

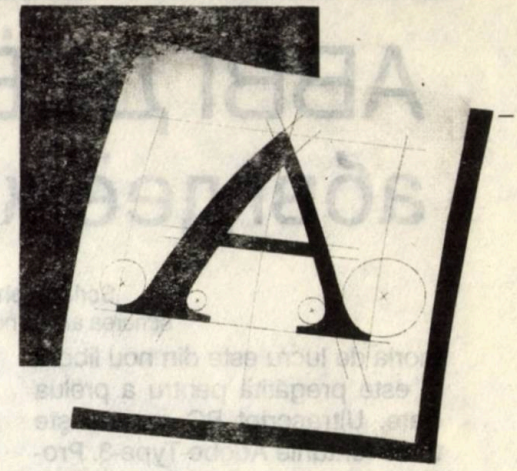
O soluţie interesantă este oferită de emulatoarele Postscript - programe care mai întîi convertesc fişierele de tipărit obţinute cu procesoare de texte sau cu programe DTP în fişiere Postscript şi abia apoi trimit aceste fişiere la imprimantă. Dezavantajul acestor interpretoare: pentru a putea imprima un fişier trebuie părăsit mai întîi programul de aplicaţie, lansat în execuţie emulatorul şi abia apoi se aşteaptă pentru imprimare. Dacă listarea trebuie repetată dintr-un motiv oarecare, întreaga procedură trebuie repetată ceea ce implică un consum ridicat de timp. Un alt dezavantaj: aceste interpretoare sînt destul de lente.

Avantajul hotărîtor: aceste emulatoare vă scutesc de o cheltuială semnificativă, în loc să aveţi nevoie de o imprimantă Postscript puteţi utiliza o imprimantă matricelă. Pentru calculatoarele MS-DOS sînt oferite trei emulatoare interesante (v. şi "if" nr. 3/90).

Freedom of Press

Translatorul Postscript al firmei Macrotron se caracterizează printr-o operare simplă. Freedom of Press calculează şi imprimă un fişier într-un ritm vioi. Se poate întîmpla însă că anumite elemente ale unei pagini să fie jertfite vitezei ridicate. Emulatorul Postscript recunoaşte toate seturile de caractere de literă Adobe-Type-1 şi Adobe-Type-3. În total programul conţine 35 de fonturi diferite. Freedom of Press poate fi utilizat pe orice calculator compatibil IBM, XT sau AT cu o memorie minimă de 640 KByte şi cu un harddisk pe care să fie liberi cel puţin 3 MByte.

Ofertant: Macrotron, 8000 München.
Preţ: cca. 800 DM.
Recomandat pentru: toate seturile de fonturi Adobe-Type-1 şi Adobe-Type-3.
Goscript 3.0 Plus



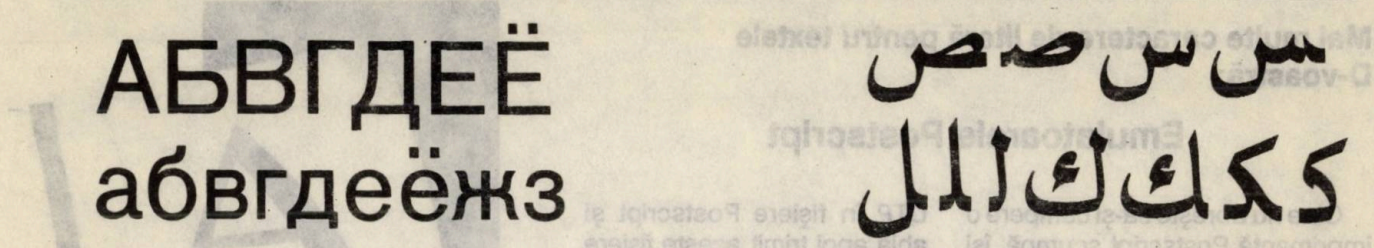
Din SUA provine "Goscript Plus" ajuns la versiunea 3.0 are nevoie de cel puţin 1 MByte de memorie şi de cel puţin 3 MByte pe harddisk.

Acest program este uşor de operat. Din păcate Goscript Plus nu poate trata decît un document la un moment dat. Imprimarea mai multor pagini cere multă răbdare din partea utilizatorului, deoarece interpretorul nu poate lista din interiorul programului de aplicaţie. Sînt recunoscute toate fonturile din setul Adobe-Type-3.

Ofertant: Michael Binkert Software,
7892 Albbruck
Preţ: cca. 550 DM.
Recomandat pentru: toate seturile de fonturi Adobe-Type-3.

Ultrascript PC 2.1

Acest translator Postscript foarte interesant este oferit de către producătorul american QMS. Programul dispune de cele mai puternice caracteristici în acest domeniu. Merită amintite mai ales posibilitatea de a imprima direct de sub un program de aplicaţie şi faptul că Ultrascript PC poate prelucra simultan mai multe pagini. Ultrascript PC poate fi încărcat ca program rezident în memorie la pornirea calculatorului. Programul poate verifica, în timpul calculului, dacă are suficientă memorie la dispoziţie. Dacă nu fişierul este redirecţionat pentru stocare temporară pe harddisk, (spooling). Datele sînt păstrate acolo într-o coadă de aşteptare pînă cînd me-



Scrierea chirilică poate fi obţinută şi cu programe shareware, scrierea arabă nu poate fi obţinută decît cu procesoare de texte speciale

moria de lucru este din nou liberă şi este pregătită pentru a prelua date. Ultrascript PC recunoaşte toate fonturile Adobe-Type-3. Programul conţine 44 de fonturi diferite.

Cerinţe sistem: PC compatibil IBM XT/AT, cu minim 640 KByte memorie de lucru (pentru imprimante laser 1 MByte), sistem de operare MS-DOS 3.1 sau mai recent, 4 MByte liberi pe harddisk.
Ofertant: Michael Binkert Software, 7892 Albbruck
Preţ: cca. 650 DM
Recomandat pentru: toate fonturile Adobe-Type-3.

Vastprint

Vastprint este un spooler de imprimantă extrem de flexibil. El utilizează multitudinea posibilităţilor de memorare pe un PC ca şi clipboard (depozit intermediar de date). Pe lîngă memoria principală a PC-ului poate fi utilizată, la dorinţă şi memoria extinsă sau harddiskul. Vastprint poate capta fişierele trimise pentru imprimare la un anumit port de ieşire (de ex. COM1 sau PAR1) şi le poate redirecta spre o altă ieşire. Cu ajutorul unei tabele de conversie fiecare caracter dintr-o "provizie ASCII" poate fi convertit în memorie şi poate fi apelat la nevoie cu ajutorul unei combinaţii de taste.

Ofertant: Dawicontrol, 3400 Gottingen
Preţ: 140 DM
Sistem de operare: MS-DOS.

Fonturi ecran din categoria Public-Domain

Din domeniul Public-Domain este de remarcat programul shareware EGA-Font. Cu toate că acest program permite afişarea caracterelor numai pe ecranul monitorului, şi nu la imprimantă, cu ajutorul lui pot fi obţinute variante de caractere amuzante. Toate cele 255 de caractere pot fi încărcate într-o mască de prelucrare şi pot fi modificate după dorinţă. Ca urmare se poate obţine, de exemplu, un Norton Commander cu caractere baroce sau o suprafaţă DOS futuristă.

Fonturi pentru Mac

Programe auxiliare: Adobe Type Manager (Apple Macintosh)

Multe programe se împăunează cu prezentarea WYSIWYG (What You See Is You Get), dar puţine îşi respectă promisiunile făcute.

Rareori o scriere este identică ca reprezentare şi pe monitor şi la imprimantă. Mai ales la caracterele de dimensiuni mari, la rotunjiri pot apare trepte urîte.

Adobe Type Manager este un program care lucrează în fundal şi care se îngrijeşte de atenuarea efectului de treaptă la reprezentarea caracterelor pe ecran.

Type Manager este livrat cu 13 fonturi Adobe, de la Times şi Helvetica şi pînă la Courier şi Symbol. Toate aceste fonturi sînt disponibile ca şi caractere îngroşate, cursive sau ca şi combinaţii de îngroşat şi cursiv.

Adobe Type Manager poate fi utilizat cu toate programele de aplicaţie uzuale, de la procesoare de texte şi pînă la DTP. De asemenea cu ajutorul acestui pachet se obţin imprimări de foarte bună calitate atît pe Hewlett-Packards Deskwriter, cît şi pe Imagewriter şi pe Laserwriter.

Adobe Type Manager este un program indicat tuturor utilizatorilor de calculatoare Mac care sînt pretenţioşi în ceea ce priveşte tehnoredactarea unor documente.

Ofertant: Adobe, Amsterdam
Preţ: 300 DM
Sistem de operare: Apple/Macintosh
Recomandat pentru: toate programele utilizator în care poate fi editat text (procesare de texte, DTP, calcul tabelar, etc.).

TypeStyler şi LetraStudio

sînt două programe puternice pentru producerea de titluri mari, antete, titluri şi logo-uri. Ele permit lucrul cu culori, rastre şi efecte speciale. Fonturile pot fi preluate din bibliotecile uzuale de fonturi, cum ar fi: The Font Company, Casady & Greeve, Bitstream, Double Click sau Adobe.

Scrierile pot fi scalate într-un domeniu larg. Extrasele de text pot fi rotite, întinse, tasate, deformate, oglindite şi remodelate. Lăţimea caracterelor ca şi distanţa dintre caractere pot fi modificate de utilizator.

Toate fonturile pot fi variate în continuare, adăugîndu-li-se o culoare, rastru, contururi etc. Scrierile pot fi aşezate flexibil pe toate formele posibile. Este posibil, de exemplu, să se îndoaie un text în jurul unei cutii sau să fie moldelat pentru gîtul unei sticle. Paleta posibilităţilor fiind atît de bogată este

destul de greu să ai o privire de ansamblu asupra întregului.

Rezultatul prelucrărilor poate fi memorat în format "EPSF" şi "PICT". În timp ce Type Styler este strîns legat de QuarkXPress, Pagemaker şi Drawing Table, LetraStudio este dedicat unei întregi serii de produse de tehnoredactare, cum ar fi: FontStudio şi DesignStudio pentru prelucrarea la nivel de pagină, ImageStudio pentru prelucrarea de imagini halftone şi ColorStudio pentru manipularea de imagini color.

Letraset Studio
Ofertant: Letraset 6000 Frankfurt
Preţ: cca. 2200 DM
Sistem de operare: Apple/Macintosh
Recomandat pentru: DesignStudio, ImagineStudio, ColorStudio şi alte programe de tehnoredactare.
Type Styler
Ofertant: Prisma, 2000 Hamburg
Preţ: cca. 800 DM
Sistem de operare: Apple/Macintosh
Recomandat pentru: QuarkXPress, Pagemaker, Drawing Table

Image Club Typeface Library

Acest program conţine o bibliotecă bogată de fonturi pentru Apple/Macintosh. Paleta se întinde de la Helvetica şi Times Roman şi pînă la caracterele de literă utilizate în germana veche, şi anume Fraktur.

"Image Club Typeface Library" este una din multele biblioteci de fonturi care pot fi utilizate de către utilizatorii de Mac. Concurenţa este constituită din Adobe, Bitstream şi alţi producători. Typeface Library oferă o selecţie uriaşă de fonturi, care sînt reproduse într-un catalog de aproape 80 de pagini. Selecţia unui font din cele mai bine de 200 de fonturi prezentate este astfel uşurată.

Fonturile oferite au nume sonore şi uneori comice, cum ar fi: "Century Schoolbook", "ITC Cheltenham" sau "Harry" şi "Rock-A-Billy".

Bibliotecile de fonturi ca Typeface Library sînt ideale pentru toţi cei care caută caractere de literă speciale pentru anumite scopuri.

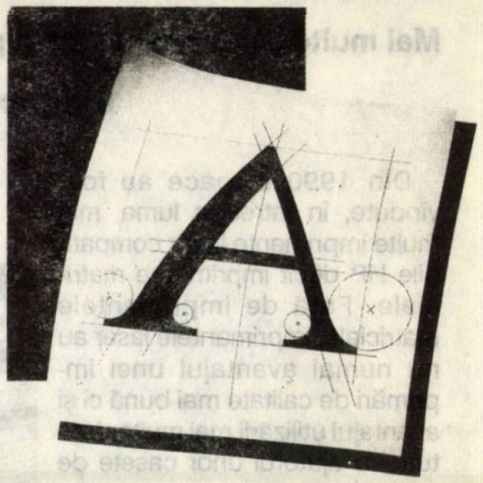
Image Club Typeface Library
Ofertant: Kagema, 7000 Stuttgart
Preţ: cca. 2000 DM
Sistem de operare: Apple Macintosh
Recomandat pentru: toate procesoarele de texte.

Fontstudio 2.0

Dacă nu se doreşte alegerea niciunuia dintre fonturile existente în bibliotecile de fonturi disponibile, atunci se poate utiliza "Fontstudio". Cu ajutorul acestui program utilizatorul îşi poate crea fonturi proprii, sau le poate remodela pe cele existente. Caracterele pot fi prelucrate fie ca bitmap fie ca outline. Astfel un caracter poate fi creat punct cu punct (bitmap), sau cu ajutorul aşa-numitelor curbe Bezier (outlines). Fontstudio permite şi cuplarea celor două moduri de lucru. Bucăţile de literă care ar putea fi reutilizate ulterior pot fi stocate într-o bibliotecă. Astfel pot fi arhivate serifele uzuale şi crochiurile de linii rezultînd un set de "scule" personal. Ca unelte de lucru sînt disponibile creionul colorat, radiera, unghiul drept, ovale şi pîlnia de umplere (desigur soft). Literele, sau anumite bucăţi din ele, pot fi rotite, scalate, oglindite sau înclinate. Caracterele individuale pot fi umbrite cu pînă la 256 de nivele de gri, putînd fi utilizate pînă la 256 de culori. Construcţia poate fi pornită de la un desen în format "PICT", "Paint" sau "TIFF".

Fontul obţinut este în format "Illustrator" şi este disponibil astfel pentru multe programe.

Fontstudio 2.0
Ofertant: Letraset, 6000 Frankfurt



Preţ: cca. 2200 DM
Sistem de operare: Apple/Macintosh
Recomandat pentru: DTP şi procesoare de texte

TScript

TScript este un interpretor Postscript. Majoritatea programelor Mac, cum ar fi: Illustrator, Freehand, Pagemaker, pot scrie comenzile Postscript şi într-un fişier şi nu trebuie trimise direct la imprimantă.

Listarea este mijlocită ulterior de programe ca TScript care ştiu interpreta comenzile Postscript. Calitatea Postscript este posibilă astfel şi cu imprimante ieftine. TScript poate converti comenzile pentru HP Laserjet şi Deskjet, respectiv Deskwriter, ca şi pentru Apple Laser şi Image Writer. Aceste imprimante devin astfel compatibile Adobe Postscript.

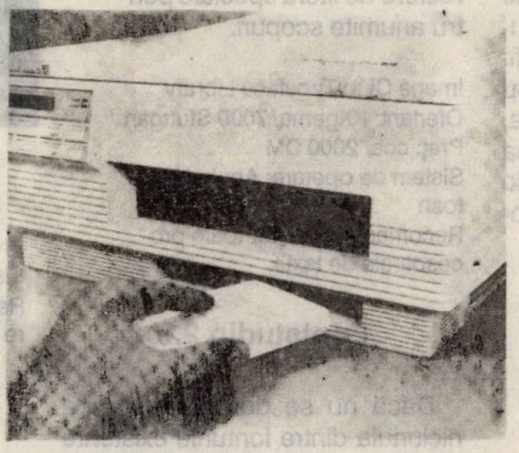
TScript, care anterior a fost comercializat sub numele "MacRIP", permite şi editarea fişierelor Postscript. Întru-cît Postscript este un limbaj de programare, utilizatorii versaţi pot prelucra documentele şi manual. Rezultatul prelucrărilor poate fi privit înainte de imprimare într-o fereastră de previzualizare.

TScript
Ofertant: Zero One, 8600 Bamberg
Preţ: cca. 450 DM
Sistem de operare: Apple/Macintosh
Recomandat pentru: DTP şi procesoare de texte

Mai multe caractere de literă pentru textele D-voastră:

6 din cele mai importante soluţii hard

Din 1990 încoace au fost vîndute, în întreaga lume, mai multe imprimante laser compatibile HP decît imprimante matriciale. Faţă de imprimantele matriciale, imprimantele laser au nu numai avantajul unei imprimări de calitate mai bună ci şi avantajul utilizării mai multor fonturi, cu ajutorul unor casete de fonturi speciale, aşa-numite "cartridges". Casetele extind nu numai paleta de fonturi ci oferă şi caractere speciale, simboluri grafice şi logo-uri. Unele casete transformă chiar o imprimantă convenţională într-o imprimantă Postscript. Pentru a putea utiliza o casetă Postscript trebuie extinsă mai întîi memoria imprimantei la cel puţin 2 MByte.



Casetele se diferenţiază între ele şi prin viteza de lucru şi prin paleta de fonturi oferite. La testarea unui produs sînt folosite ca referinţă casetele firmelor americane Pacific Page sau IQ-Engineering.

Puteţi cumpăra următoarele casete de fonturi:

Hewlett-Packard-International-Collection

Caseta este recomandată pentru toate modelele HP Laserjet II şi Lasejet IID/IIP. Caseta conţine trei fonturi (Letter Gothic, Prestige Elite şi Courier; fiecare în forma normală, îngroşată şi cursivă), cu distanţe fixe între caractere.În plus mai sînt conţinute CG Times şi Univers Normal (două scrieri proporţionale). Cu Roman-8, PC-8, PC-D/N şi PC 850 sînt puse la dispoziţie patru seturi de caractere. Caseta poate fi utilizată şi de modelele mai vechi 500+ şi Laserjet 2000.

Ofertant: Macrotron, 8000 München
Preţ: cca. 780 DM

Hewlett-Packard-Cartridge

Caseta este destinată modelelor HP Laserjet IID/P şi Laserjet III şi conţine 35 de fonturi standard Postscript ale specialistului în fonturi Adobe. Pentru utilizarea casetei este necesară o memorie de cel puţin 2 MByte.

Ofertant: Macrotron, 8000 München
Preţ: cca. 1700 DM

Pacific-Page / Personal-Edition (PE)

Casetă de fonturi pentru HP Laserjet IIP/D, Laserjet III şi Siemens PT-7500/ PT-7800. Caseta conţine 35 de fonturi Adobe Postscript. Pentru a putea fi utilizată este necesară o memorie de cel puţin 1,5 MByte. Dacă imprimanta dispune de o memorie de cel puţin 3 MByte atunci se lansează aşa-numita Double-Buffering, ceea ce determină o sporire a vitezei de lucru.

Ofertant: AMS Computech,
8000 München
Preţ: cca. 1700 DM

Pacific-Page /25 in One

Casetă de fonturi pentru Laserjet II, Laserjet IIP/D, Laserjet III, Laserjet IIID, Canon LBP8-II, Siemens PT-7500 şi Siemens PT-7800. Caseta conţine întreaga paletă de fonturi a 25 de casete HP diferite, în total 172 de fonturi. Dimensiunile de caractere posibile se găsesc în plaja de valori de la 4 pînă la 18 puncte. Nu este necesară o extensie de memorie.

Ofertant: AMS Computech,
8000 München
Preţ: cca. 750 DM

Super-Cartridges 2

Casetă de fonturi pentru HP Laserjet II, IIP/D, III. Caseta conţine 17 scrieri diferite şi 8 seturi de caractere cu dimensiuni între 4 şi 30 de puncte. Nu este necesară o extensie de memorie. Furnitura de livrare conţine şi drivere pentru diferite procesoare de texte şi Windows 3.0

Ofertant: Computer Connections,
4030 Ratingen-Lintorf
Preţ: cca. 1130 DM

Super-Cartridges 3

Casetă de fonturi pentru HP Laserjet III. Caseta conţine 44 de fonturi Postscript. Toate fonturile pot fi imprimate la dimensiuni cuprinse între 3,5 şi 30 de puncte. Nu necesită memorie suplimentară.

Ofertant: Computer Connection,
4030 Ratingen-Lintorf
Preţ: cca. 1470 DM

(R.M.)

Memento: Comenzi Paradox 3.5

=	atribuie o valoare unei variabile, tablou sau cîmp NumeVariabilă = expresie NumeTablou [Element] = expresie SpeçCîmp = expresie
?	scrie în ecranul utilizator (canvas) PAL pe linia care urmeaz după cursor ? [ListăExpresii]
??	scrie în ecranul utilizator PAL pe linia pe care se află cursorul ?? [ListăExpresii]
@	poziţionează cursorul în ecranul utlizator @ Linie,Coloană
ACCEPT	acceptă o valoarea de la tastatură ACCEPT TipDată [PICTURE format] [MIN valoare] [MAX valoare] [LOOKUP NumeTabelă] [DEFAULT valoare] [REQUIRED] TO NumeVariabilă
ADD	adaugă articolele dintr-o tabelă la cele din altă tabelă ADD NumeTabelă1 NumeTabelă2
ARRAY	defineşte un tablou ARRAY NumeTablou [Număr]
BACKSPACE	şterge cararacterul aflat la stînga cursorului BACKSPACE
BEEP	emite un semnal sonor (beep) BEEP
CALCDEBUG	comută afişarea mesajelor pentru cîmpurile calculate on sau off; setează şirul de atribuire pentru date invalide CALCDEBUG { ON | OFF } Şir
CANCELEDIT	întrerupe o operaţie CANCELEDIT
CANVAS	controlează afişarea pe ecranul utilizator PAL (canvas) CANVAS [ON | OFF]
CHECK	<F6> plasează sau şterge un marcaj check într-o cerere de interogare CHECK
CHECK-DESCENDING	<Ctrl/F6> plasează sau şterge un marcaj ceck descending într-o cerere de interogare CHECKDESCENDING
CHECKPLUS	<Alt/F6> plasează sau şterge un marcaj checkplus într-o cerere de interogare CHECKPLUS
CLEAR	şterge ecranul utilizator PAL CLEAR [{ EOL | EOS }]
CLEARALL	<Alt/F8> şterge toate imaginile din spaţiul de lucru CLEARALL
CLEARIMAGE	<F8> şterge imaginea curentă din spaţiul de lucru CLEARIMAGE
CLOSE PRINTER	închide accesul la o imprimantă CLOSE PRINTER
COEDIT	permite coeditarea unei tabele COEDIT NumeTabelă
COEDITKEY	<Alt/F9> stabileşte modul de lucru coedit COEDITKEY
COPY	copiază o tabelă şi familia ei COPY NumeTabelă1 NumeTabelă2
COPYFORM	copiază o formă COPYFORM NumeTabelă1 NumeFormă1 NumeTabelă2 NumeFormă2
COPYFROM-ARRAY	copiază elementele dintr-un tablou într-o înregistrare a unei tabele COPYFROMARRAY NumeTablou
COPYREPORT	copiază un raport COPYREPORT NumeTabelă1 NumeRaport1 NumeTabelă2 NumeRaport2
COPYTO-ARRAY	copiază o înregistrare într-un tablou COPYTOARRAY NumeTablou
CREATE	crează o nouă tabelă CREATE NumeTabelă ListăDefCîmpuri CREATE NumeTabelă1 LIKE NumeTabelă2
CREATELIB	crează o nouă bibliotecă de proceduri CREATELIB NumeBibliotecă [SIZE Număr]
CROSS-TABKEY	<Alt/X> stabileşte referinţele încrucişate pentru tabela curentă CROSSTABKEY
CTRL-BACKSPACE	<Ctrl/Backspace> şterge cîmpul curent CTRLBACKSPACE
CTRLBREAK	<Ctrl/Break> opreşte exec. prog. curent CTRLBREAK
CTRLEND	acelaşi efect ca şi <Ctrl/End> CTRLEND
CTRLHOME	acelaşi efect ca şi <Ctrl/Home> CTRLHOME
CTRLLEFT	acelaşi efect ca şi <Ctrl/←> CTRLLEFT
CTRLPGDN	acelaşi efect ca şi <Ctrl/PgDn> CTRLPGDN
CTRLPGUP	acelaşi efect ca şi <Ctrl/PgUp> CTRLPGUP
CTRLRIGHT	acelaşi efect ca şi <Ctrl/→> CTRLRIGHT
CURSOR	comută afişarea cursorului pe ecanul utilizator on / off, sau selectează forma cursorului CURSOR {BAR | BOX | NORMAL | OFF}
DEBUG	opreşte execuţia unui program (script) şi porneşte depanatorul DEBUG
DEL	acelaşi efect ca şi <Del> DEL
DELETE	şterge o tabelă şi familia sa DELETE NumeTabelă
DELETELINE	<Ctrl/Y> şterge începînd de la cursor pînă la sfîrşitul liniei DELETELINE
DITTO	<Ctrl/D> repetă valoarea cîmpului din înregistrarea precedentă DITTO

CONSECO S.R.L.

cu sediul în Baia Mare , C.P.517, str. Siugariu 23/38, tel./fax: 40-94-36433

în colaborare cu

Societatea Comercială Româno - Canadiană

JM & CO. S.R.L.

din Bucureşti, str. Miletin nr. 1, sectorul 3, tel.:20.33.24, tel./fax : 20.27.05

vă oferă :

A) REŢELE LOCALE ARTISOFT " LANtastic "

- **LANtastic AE-2 Ethernet Starter Kit (2 adaptoare de reţea, soft pentru maxim 300 utilizatori, ca blu coaxial subţire mufat de 8 m, 2 conectori T, 2 terminatori, documentaţii pentru hard şi soft) - magistrala IDE. 875 $**
- **LANtastic 2MBps Starter Kit (2 adaptoare de reţea, soft pentru maxim 300 utilizatori, cablu torsadat mufat de 5 m, 2 conectori T, 2 terminatori, documentaţii pentru hard şi soft) - magistrala IDE. 625 $**
- **LANtastic Voice Adapter Kit (inclusiv receptor telefonic). 124 $**
- **LANtastic NETWORK EYE (program pentru supravegherea activităţii reţelei şi modul ei de funcţionare) 225 $/reţea**

De asemenea, vă putem oferi toate componentele hard şi soft necesare pentru orice configuraţie dorită.

Pentru celelalte preţuri vă rugăm să ne contactaţi.

GARANŢIA OFERITĂ PENTRU TOATE PRODUSELE ARTISOFT ESTE DE 5 ANI.

B) Dispozitive de tuşat benzi pentru imprimante de calculator.

Dispozitivul electric semiautomatizat de tuşat, inclusiv o sticluţă de tuş, costă **49000 lei**, *iar rezervele suplimentare* **4700 lei/buc**. *Cu o rezervă de tuş se pot tuşa cca. 30 de cartuşe, de orice lăţime - şi orice tip de cartuş, între 6 - 30 mm fără a necesita scoaterea benzii din cartuş.*

C) GOALKEEPER - o alternativă superioară la U.P.S.

- **la o cădere de tensiune între 0 - 3 sec. GOALKEEPER suplineşte instantaneu sursa de curent continuu a calculatorului.**
- **la o cădere de tensiune între 3 - 90 sec.- GOALKEEPER face back-up în 10 - 90 sec., depinzînd de mărimea memoriei RAM. Dacă între timp revine tensiunea, calculatorul nu trebuie oprit, putîndu-se continua lucrul.**
- **la o cădere de tensiune mai mare de 90 sec. - GOALKEEPER salvează toate datele pe harddisk şi opreşte calculatorul. Cînd revine tensiunea, se pot restaura toate datele.**

Cerinţe hardware şi software: un calculator IBM PC, XT sau AT sua compatibil, harddisk, DOS 2.1 sau versiuni mai noi , cel puţin 256 Kb RAM.

Dimensiuni : 21.5 x 14.5 x 4.7 cm
Greutate : 3 Kg
Preţul : **69500** *LEI*

MOVETO mută cursorul la un cîmp, înregistrare sau imagine specificată
MOVETO SpecificatorCîmp
MOVETO RECORD Număr
MOVETO Număr
MOVETO Imagine
MOVETO FIELD NumeCîmp

OPEN PRINTER deschide accesul la imprimantă
OPEN PRINTER

PAINTCANVAS stabileşte atributele ecranului utilizator
PAINTCANVAS [BORDER] [FILL Şir] {ListăOpţiuni | ATTRIBUTE Număr | BACKGROUND} Lin1, Col1, Lin2, Col2

PASSWORD stabileşte accesul la o tabelă protejată
PASSWORD ListăParole

PGDN acelaşi efect ca şi <PgDn>
PGDN

PGUP acelaşi efect ca şi <PgUp>
PGUP

PICKFORM selectează o formă
PICKFORM NumeFormă

PLAY execută un program (script)
PLAY NumeProgram

PRINT listează valorile specificate la imprimantă sau la într-un fişier
PRINT ListăExpresii
PRINT FILE NumeFişier ListăExpresii

PRINTER trimite toate ieşirile la imprimantă şi pe ecran
PRINTER { ON | OFF }

PRIVTABLES memorează tabelele în directoare private
PRIVATE ListăTabele

PROC defineşte o procedură PAL
PROC [CLOSED] NumeProcedură ([ListăNumeVariabile1])
[USEVARS ListăNumeVariabile2]
[PRIVATE ListăNumeVariabile3]
Comenzi
ENDPROC

PROMPT înlocuieşte prompter-ul Paradox din primele două linii
PROMPT [Prompter1, Prompter2]

PROTECT protejează o tabelă prin parolă
PROTECT NumeTabelă Parolă

QUERY plasează o cerere în spaţiul de lucru
QUERY
[{ TABLEORDER | IMAGEORDER }]
FormeCerere.
ENDQUERY

QUIT termină execuţia tuturor programelor şi se revine în Paradox
QUIT [Expresie]

QUITLOOP ieşire dintr-o buclă
QUITLOOP

READLIB extrage proceduri dintr-o bibliotecă
READLIB NumeBibliotecă [IMMEDIATE] ListăNumeProceduri

REFRESH <Alt/R> reîmprospătează imaginea de pe ecran

RELEASE elimină proceduri, variabile, sau tablouri din memorie
RELEASE PROCS { ALL | ListăNumeProc}
RELEASE VARS { ALL | ListăNumeVar }

RENAME redenumeşte o tabelă şi familia acesteia
RENAME NumeTabelă1 NumeTabelă2

REPORT imprimă un raport
REPORT NumeTabelă NumeRaport

REPORT-TABLES listează tabelele înlănţuite într-un raport multitabelar
REPORTTABLES

REQUIRED-CHECK comută on/off procedura de validare pentru cîmpurile interogate
REQUIREDCHECK { ON | OFF }

RESET revenire la meniul principal Paradox
RESET

RESYNCKEY <Ctrl/L> resincronizează înregistrările într-o formă multitabelară
RESYNCKEY

RETURN revenire dintr-un program (script) sau procedură
RETURN [Expresie]

REVERSETAB <Shift/Tab> mută cursorul la cîmpul anterior
REVERSETAB

RIGHT acelaşi efect ca şi <→>
RIGHT

ROTATE <Ctrl/R> roteşte cîmpurile aflate la dreapta cursorului
ROTATE

RUN execută o comandă sau un program DOS
RUN [BIG] [SLEEP Număr]
[NOREFRESH] [NORESTORE]
[NOSHELL] ComandăDOS

SAVETABLES scrie buffer-ele tabelelor pe disc
SAVETABLES

SAVEVARS salvează valorile variabilelor şi tablourilor într-un fişier
SAVEVARS { ALL | ListăNumeVariabile}

SCAN parcurge o tabelă înregistrare cu înregistrare
SCAN [FOR Condiţie]
[Comenzi]
ENDSCAN

SELECT selectează o comandă sau tastează un răspuns la prompter-ul unui meniu
SELECT Expresie

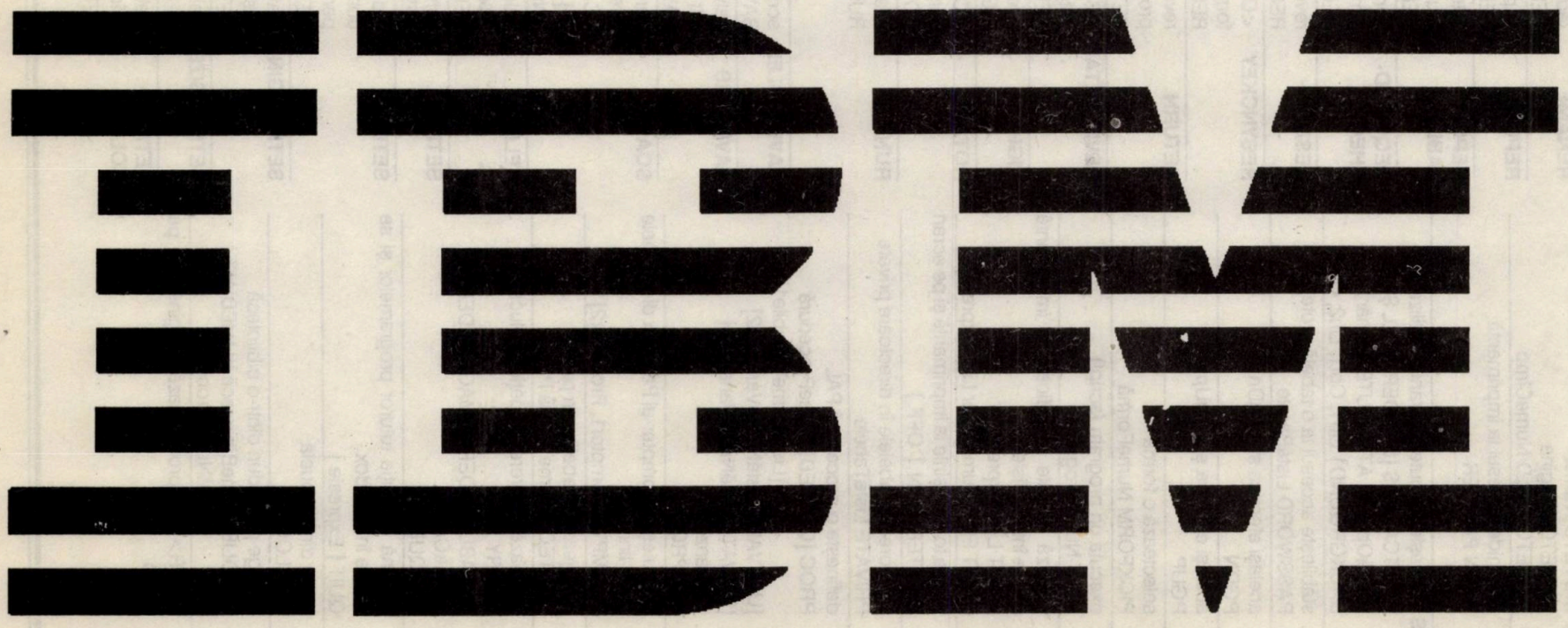
SETDIR schimbă unitatea sau directorul de lucru
SETDIR CaleDOS

SETKEY redefineşte operaţia executată de o combinaţie de taste (defineşte un macro pentru tastatură)
SETKEY CodTastă [Comenzi]

SETMARGIN stabileşte marginea din stînga pentru textul afişat
SETMARGIN { OFF | Număr }

SETMAXSIZE stabileşte dimens. maximă a unei tabele
SETMAXSIZE Număr

SETNEG-COLOR stabileşte culoarea care va fi folosită pentru evidenţierea valorilor negative dintr-o tabelă
SETNEGCOLOR {CURRENCY | NUMERIC | BOTH} { ON | OFF }



micro ATCI S.R.L.

str. Gheorghe Doja nr. 36
C.P. 64 O.P. 1
4300 Tîrgu Mureş
Telefon/Telefax: 954/31660

vă oferă întreaga gamă
de produse originale

DO_IT! <F2> termină sau execută o operaţie
DO_IT!

DOS <Ctrl/O> suspendă Paradox şi accede sistemul de operare DOS
DOS

DOSBIG <Alt/O> suspendă Paradox şi accede S.O. DOS cu max. de memorie RAM disponibilă
DOSBIG

DOWN acelaşi efect ca şi < ↓ >
DOWN

DOWNIMAGE <F4> mută în jos o imagine în spaţiul de lucru sau într-o formă multitabelară
DOWNIMAGE

ECHO afişează modul de lucru intern al unui program (script)
ECHO {FAST | SLOW | OFF | NORMAL}

EDIT permite editarea unei tabele
EDIT NumeTabelă

EDITKEY <F9> stabileşte modul de lucru Edit
EDITKEY

EDITLOG acceptă sau reface modificările făcute într-o tabelă aşa cum au fost înregistrate în jurnalul tranzacţiilor
EDITLOG {MARK | REVERT | INDEX | PERMANENT}

EMPTY şterge toate înregistrările dintr-o tabelă
EMPTY NumeTabelă

END acelaşi efect ca şi <End>
END

ENTER acelaşi efect ca şi <Enter>
ENTER

ESC <Esc> revenire la meniul anterior, prompter sau spaţiul de lucru
ESC

EXAMPLE <F5> introduce un exemplu pentru o cerere
EXAMPLE

EXECPROC execută o procedură fără argumente
EXECPROC NumeProcedură

EXECUTE execută o secvenţă de comenzi PAL
EXECUTE Comenzi

EXIT părăseşte Paradox
EXIT

FIELDVIEW <Alt/F5> sau <Ctrl/F> stabileşte modul de lucru editare cîmp (field view)
FIELDVIEW

FIRSTSHOW face ca imaginea curentă să fie prima imagine din spaţiul de lucru
FIRSTSHOW

FOR controlează de cîte ori va fi executată o secvenţă de instrucţiuni
FOR NumeVariabilă [FROM Număr1]
[TO Număr2] [STEP Număr3]
Comenzi
ENDFOR

FORMKEY <F7> comută între modurile de afişare tabelar şi formă
FORMKEY

FORMTABLES listează tabelele îmbricate într-o formă multitabelară
FORMTABLES NumeTabelă NumeFormă NumeTablou

GRAPHKEY <Ctrl/F7> reprezintă grafic tabela curentă
GRAPHKEY

GROUPBY <Shift/F6> grupează articolele pentru cererea curentă fără să le afişeze
GROPUPBY

HELP <F1> afişează ecrane de ajutor
HELP

HOME acelaşi efect ca şi tasta <Home>
HOME

IF execută anumite comenzi în funcţie de îndeplinerea unor condiţii
IF Condiţii
THEN Comenzi1
[ELSE Comenzi2]
ENDIF

IMAGERIGHTS setează sau şterge drepturile de acces la o tabelă
IMAGERIGHTS [{UPDATE | READONLY}]

INDEX crează un index secundar pentru o tabelă
INDEX [MAINTAINED] NumeTabelă
ON NumeCîmp

INFOLIB afişează numele de proceduri dintr-o bibliotecă
INFOLIB NumeBibliotecă

INS acelaşi efect ca şi <Ins>
INS

INSTANTPLAY <Alt/F4> execută programul Instant
INSTANTPLAY

INSTANT-REPORT <Alt/F7> listează la imprimantă raportul standard pentru tabela curentă
INSTANTREPORT

KEYLOOKUP <Alt/K> afişează înregistrările cu chei duplicate
KEYLOOKUP

KEYPRESS transmite orice caracter spre Paradox ca şi cum ar fi fost tastat
KEYPRESS CodTastă

LEFT acelaşi efect ca şi <←>
LEFT

LOCATE localizează o înregistrare într-o tabelă
LOCATE [NEXT] [PATTERN] ValoareCîmp
LOCATE [NEXT] ListăValoriCîmp

LOCK blochează accesul la una sau mai multe tabele
LOCK ListăBlocare

LOCKKEY <Alt/L> blochează sau deblochează articolul curent
LOCKKEY

LOCKRECORD blochează o înregistrare
LOCKRECORD

LOOP revenire la începutul unei bucle
LOOP

MENU <F10> afişare meniu
MENU

MESSAGE afişează un mesaj
MESSAGE ListăExpresii

Cine ridică mănuşa?

În numărul din septembrie 91 al revistei Dr. Dobbs' Journal Ray Valdés, într-un articol intitulat "Little Languages - Big Questions", lansează o provocare.

Din punct de vedere "calculatoristic" (computationally speaking), unele limbaje mici sînt foarte "puternice", în ciuda aparenţei lor insignifiante, iar altele sînt nişte "piticoţi" (travestiţi în giganţi).

Deşi există teste care să permită evaluarea codului generat de un compilator sau a vitezei de compilare, nu există încă teste (benchmarks) care să permită evaluarea "puterii" unui limbaj de programare - a posibilităţilor de a exprima în el anumite lucruri. Într-un sens mai larg, toate limbajele - pînă la cele mai rudimentare - sînt echivalente, pentru că toate sînt capabile să acţioneze ca maşini Turing. Privite în detaliu, totuşi, limbajele sînt suficient de diferite - o comparaţie fiind posibilă la fel ca o comparaţie între mere şi portocale.

În ciuda acestei observaţii, expresivitatea unui limbaj poate fi măsurată - prin uşurinţa cu care pot fi exprimate probleme fundamentale.

Un test, mai elocvent, este exerciţiul de programare cu care Ken Thompson (coautor al Unix-ului) îşi trecea timpul în studenţie. Conform unei declaraţii a lui (1983, la primirea premiului ACM Turing) - exerciţiul ce urmează este popular "din acelaşi motiv pentru care sînt populare speciile de animale cu 3 picioare". Scopul este de a scrie cel mai scurt program autoreproducător: "mai exact, de a scrie un program sursă care, după compilare şi execuţie, va produce ca ieşire o copie exactă a sursei sale".

Thompson adaugă: "dacă nu aţi făcut niciodată acest lucru, vă sugerez cu insistenţă să încercaţi singur. Descoperirea soluţiei este o revelaţie care depăşeşte cu mult orice beneficiu obţinut prin faptul că cineva vă dă soluţia de-a gata".

Programul lui Thompson, în C, este prezentat în Exemplul 1. Deşi el îl consideră ca punct de plecare pentru "cel mai isteţ program pe care l-am scris vreodată", codul constă din 223 linii de C oarecum banal.

Prin comparaţie, echivalentul acestui program poate fi exprimat, foarte elegant, în Lisp pur, în numai 3 linii de cod (ex. 2). (Lisp-ul pur este un subset al Lisp-ului constînd din funcţii care nu au efecte colaterale, deci funcţii care din punct de vedere matematic sînt "pure". Codul din Ex. 2 este scris de John Mc Carthy şi Carolyn Talcott).

În exemplul 3, este prezentat un program echivalent în Mumps - un limbaj asemănător Basic-ului, care accede o sotisficată maşină - bază - de date - asociativă (associative database engine). Deşi listingul are numai cîteva linii, codul este extrem de enigmatic şi criptic.

Cu toate că fiecare dintre aceste programe are la bază aceeaşi logică, paşii necesari ajungerii la ţintă, în fiecare limbaj, sînt atît de diferiţi încît s-ar putea paria că fiecare program implementează un algoritm distinct.

Cum se descurcă limbajul Dvs. preferat (sau cel mai nesuferit)? Soluţiile Dvs. pentru acest exerciţiu sînt aşteptate (atît de noi, cît şi) de Dr. Dobb's Journal, M&T Publishing Inc., 501 Galveston Dr., Redwood City, CA 94063, for Ray Valdés. Listingurile cele mai interesante, elegante, şi/sau bizare vor fi publicate. Un premiu special va fi acordat primei soluţii scrisă în limbajul de macrouri propriu Lotus-ului.

(I.F.)

Ex.1. C

```
char s[] = {
    '\t',
    '0',
    '\n',
    '}',
    ';',
    '\n',
    '/',
    '*',
    '\n',
    ... 213 lines deleted ...
    0
};
/*
 * Sirul s este o reprezentare a acestui program,
 * de la '0' pina la end.
 */
main ()
{
    int   i;

    printf ("char\ts[] = {\n");
    for (i=0; s[i]; i++)
        printf ("\t%d, \n", s[i]);
    printf ("%s", s);
}
```

Ex. 2 Lisp pur

```
( (lambda (x) (list x (list (quote quote) x)))
    (quote
        (lambda (x) (list x (list (quote quote) x ))) ))
```

Ex. 3 Mumps

```
S N=$C(13)_$C(10)_$C(9) S Q=$C(34) S S="W
N_""S N=$C(13)_$C(10)_$C(9) S Q=$C(34) S
S=""_Q;F I=1:1:$L(S) S C=$E(S,I) W
$S(C=$C(34):Q_Q,1:C);W Q_N;F I=1:1:$L(S) S
C=$E(S,I) W $S(C=$C(59):N,1:C);Q;"
W N_""S N=$C(13)_$C(10)_$C(9) S Q=$C(34) S
S="_Q
F I=1:1:$L(S) S C=$E(S,I) W $S(C=$C(34):Q_Q,1:C)
W Q_N
F I=1:1:$L(S) S C=$E(S,I) W $S(C=$C(59):N,1:C)
Q;
```

Care credeţi că este cel mai folosit limbaj de programare folosit în prezent? Deşi multora dintre noi le-ar place, nici C-ul şi nici Pascal-ul nu stăpînesc lumea. E prea tîrziu pentru ADA sau ASM, şi prea devreme pentru C++. Fortran sau Cobol. Atunci? Răspunsul corect: nici unul din toate acestea.

De fapt, limbajul de programare cu cea mai largă răspîndire este limbajul de macrouri din Lotus 1-2-3! Sigur, programatorii "adevăraţi" nu folosesc macrolimbajul Lotus, dar un număr foarte mare de oameni o fac - şi obţin rezultate utile, în fiecare zi.

Mumps este un limbaj de programare ce foloseşte o sintaxă asemănătoare Basic-ului. Mumps leagă un amestec bogat de funcţionalităţi sofisticate tip bază de date de o sintaxă - schelet, constînd din 26 de comenzi alcătuite din cîte o literă a alfabetului, incluzînd "G"pentru "goto". Această ciudăţenie de limbaj este baza pentru unul din cele mai mari proiecte de dezvoltare software lansate vreodată, contractul guvernamental de 1,2 miliarde $, ce începe acum, pentru computerizarea spitalelor americane.

Turbo Vision - sau cum să moştenim roata în viziunea Borland

În primăvara acestui an profetul limbajelor de programare orientate obiect, Borland, a împărtăşit credincioşilor săi ultima sa viziune. Turbo Vision nu este doar o colecţie de obiecte ce trasează cadrul general al unei aplicaţii, ci este şi suportul unui concept nou în programarea PC-urilor. Deşi timpul de iniţiere a crescut, a scăzut dramatic durata finisării unei aplicaţii iar cizelarea ei poate fi realizată impecabil.

Ceea ce era previzibil încă din toamna lui 1988 a devenit acum evident: unificarea suprafeţelor utilizator, a interfeţelor program şi a protocoalelor de control În oferta de programe aplicative. Windows 3.0 vine în forţă iar producătorii se grăbesc să se alinieze noului standard.

Dintre producătorii de compilatoare primul care a reacţionat cu promptitudine este Borland care înaintează cu încă un pas cu ale sale Turbo Pascal 6.0 şi Turbo Pascal for Windows (TPW). Succesul nu este asigurat atît de calitatea remarcabilă a compilatoarelor în sine cît de cele două biblioteci livrate împreună cu acestea. Dacă Object Windows Library (OWL) -oferit cu TPW- furnizează un cadru general pentru o aplicaţie Windows, Turbo Vision (TV) - livrat cu TP 6.0 - oferă aceeaşi structură pentru o aplicaţie În mod text rulînd sub MS-DOS.

TV şi OWL prezintă o similitudine remarcabilă deşi implementează idei identice pentru două platforme foarte diferite (frapantă nu este atît diferenţa mod text - mod grafic, cît diferenţa mod real de adresare - mod protejat / multitasking). Amîndouă oferă o arhitectură de program bazată pe tratarea evenimentelor, împreună cu componentele soft pentru interfeţe cu utilizatorul şi respectiv cu sistemul.

Ambele sînt în mod inevitabil orientate obiect. Ideea de bază este că realizarea unei aplicaţii concrete se face prin moştenirea unui cadru general de aplicaţie, gol, la care se adaugă cod pentru realizarea sarcinilor specifice aplicaţiei. Cadrul conţine cuiele necesare atîrnării pieselor componente şi este dispus pe axele carteziene OOP ale generalităţii şi specificităţii. Preluînd comparaţia dragă autorilor lui Turbo Vision, scheletul unei aplicaţii se constituie din meniuri, ferestre, cîmpuri de editare, rutine de tratare a erorilor, de control al mouse-ului, în timp ce carnea rezolvă probleme specifice aplicaţiei. Cele mai multe aplicaţii se conformează unei asemenea reprezentări, iar programatorii scriu adesea mai multe linii de cod pentru descrierea scheletului decît pentru materializarea cărnii, nereuşind să facă o distincţie clară între ele. Acum Borland ne propune ca În loc să reinventăm roata s-o moştenim. Şi nu o roată oarecare ci mii de linii de program testat şi optimizat ce ar putea reduce propria aplicaţie la surprinzător de puţin cod.

Aici este momentul să ne separăm de OWL. Să lăsăm bufniţa să zboare, căci toate consideraţiile care urmează rămîn perfect valabile şi pentru ea, cel puţin din punct de vedere conceptual. Portabilitatea codului dinspre TV înspre OWL este remarcabilă, diferenţele ţinînd de încorporarea în Windows a unei părţi din arhitectura de control a evenimentelor, de modul specific de creare a resurselor grafice. Majoritatea obiectelor TV se regăsesc în OWL.

Abstracţiuni

Turbo Vision se prezintă cu trei categorii de obiecte, grupate pe criterii funcţionale: obiecte primordiale, obiecte vizibile pe ecran (le vom referi deseori ca *views*), precum şi obiecte invizibile.

În ultimele două categorii există numeroase tipuri de obiecte cu caracter abstract. Ele au practic potenţialul de a face orice dar rămîn prea generale pentru a face ceva concret. Utilizatorul trebuie să definească tipuri de obiecte descendenţi care să întreprindă acţiuni specifice folosind cît mai mult din codul cu caracter general al obiectului părinte. Paradoxal, tocmai existenţa acestor obiecte abstracte conferă soliditate conceptuală bibliotecii şi serveşte din plin dezideratul reducerii efortului de programare.

Să considerăm pentru exemplu trei tipuri de obiecte: *TWindow*, *TDesktop* şi *TProgram*. Acestea au în comun o caracteristică esenţială: fiecare poate conţine, (în sensul reprezentării pe ecran), mai multe obiecte vizibile (views) care se află pe acelaşi nivel al aplicaţiei şi "concertează" împreună. *TProgram* grupează un meniu-linie (*TMenuBar*), o linie status (*TStatusLine*) şi un suport - *TDesktop* - care la rîndul lui poate conţine un număr practic nelimitat de *view*-uri: boxe de dialog, cîmpuri de introducere a datelor, ferestre, etc. O fereastră (*TWindow*) conţine un cadru, bare de desfăşurare (*scrolling bars*), diverse alte obiecte: butoane, liste, ...

. Interacţiunea dintre obiectele conţinute, selectarea lor cu mouse-ul sau tastatura se face prin intermediul obiectului posesor, *TProgram*, *TDesktop* sau *TWindow*. Această importantă trăsătură, de a deţine şi coordona *view*-uri, a impus necesitatea definirii tipului abstract de obiect grup - *TGroup* - din care derivă cele trei obiecte În discuţie. Pentru programator devine banală acum derivarea din *TGroup* a unui tip original de obiect-cadru de aplicaţie, cu un comportament similar lui *TProgram* dar cu un meniu standard vertical şi o linie status situată în partea de sus a ecranului, de exemplu.

Utilitatea instanţierii unui tip de obiect depinde de circumstanţe. Numeroase tipuri au metode abstracte ce impun definirea lor în descendenţi. Există ş categoria metodelor pseudeo-abstracte care oferă un minimum de acţiuni implicite; redefinirea lor nu este necesară decît dacă nu corespund scopurilor urmărite de programator. Ca o regulă generală, pe măsură ce se coboară în ierarhia TV, de la tipurile primordiale de obiecte la descendenţii lor cei mai îndepărtaţi, obiectele devin tot mai specializate şi mai puţin abstracte, putînd fi instanţiate direct, fără a mai deriva descendenţi.

Obiecte şi poze

Revenind la categoriile de obiecte Turbo Vision remarcăm trei tipuri primitive, extrem de simple. *TPoint* şi *TRect* reprezintă un punct şi respectiv o zonă dreptunghiulară pe ecran. Fără descendenţi şi avînd puţine metode, ele sînt utilizate intens de toate tipurile de obiecte vizibile. De altfel *view*-urile sînt fără excepţie dreptunghiulare. "Tatăl tuturor", *TObject*, este un tip abstract, fără cîmpuri, cu trei metode: *Init*, baza tuturor constructorilor TV, alocă memorie şi iniţializează cu zero toate cîmpurile descendenţilor; *Free* dealocă memoria iar *Done* este un destructor abstract ce trebuie redefinit în descendenţi. *TObject* este tipul de bază din care derivă absolut toate tipurile de obiecte TV cu excepţia lui *TPoint* şi *TRect*.

Cărămida de bază a unei aplicaţii TV este obiectul vizibil - *view* - descendent al *TView*. În fapt tot ceea ce apare pe ecran **trebuie** să fie sau să aparţină unui obiect vizibil, iar proprietatea sa cea mai importantă este aceea de a şti să se autoreprezinte ori de cîte ori condiţiile o cer. View-urile pot şi trebuie să coexiste; ca urmare se pot suprapune, redimensiona, pot comunica între ele, se pot deplasa pe ecran.

Acelor cititori familiarizaţi cu Windows nu le va fi greu să sesizeze particularităţile tipurilor de obiecte vizibile ce urmează a fi trecute în revistă. De la ferestre la butoane, Turbo Vision reuşeşte o remarcabilă emulare a graficii foarte personalizate de sub Windows cu umila paletă a celor 256 de caractere IBM. Chiar dacă în ceea ce priveşte aspectul "text" ar mai fi cîte ceva de cîrtit, comportamentul view-urilor este impecabil iar timpii de afişare a imaginii pe ecran sînt insesizabili - cu toţii ne amintim cu neplăcere lenea cu care Windows pictează ecranele AT-urilor cu plăci grafice lente.

Portrete de familie

Numeroase sînt tipurile "grup" de obiecte vizibile. A programa fără ele în TV este de neconceput: aplicaţia însăşi este un grup care acoperă întregul ecran, dar aceasta nu se observă întrucît dialogul utilizatorului decurge cu obiecte specifice, administrate de aplicaţie.

Deseori, ceea ce pare a fi un singur obiect este în fapt un grup de obiecte aflate în interacţiune. Astfel se prezintă boxele de dialog - descendenţi ai lui TDialog - ce conţin "*subview*"-uri precum butoanele (*TButton*), căsuţele de bifare (*TCheckBoxes*), butoanele radio (*TRadioButtons*), cîmpuri de editare (*TInputLine*). *TDialog* este un caz mai special de fereastră, de dimensiune fixă, dar care poate fi "plimbată" pe ecran. Uneori existenţa ei se limitează strict la interacţiunea cu utilizatorul: atunci dialogul este imperativ, utilizatorul nu are posibilitatea s-o lase să aştepte şi să dialogheze cu alte obiecte. Se spune că dialogul este în stare "modală" de execuţie. Obiectele modale se diferenţiază de cele "inserate" în grup, ce pot fi selectate şi abandonate, iar distrugerea lor se face în mod explicit şi nu la abandonarea lor. De remarcat că există întotdeauna cel puţin un obiect modal într-o aplicaţie, iar acesta este aplicaţia însăşi. Indiferent că derivă din *TApplication* sau este un tip mai special derivat de programator din *TGroup*, ea este executată în stare modală în cadrul unei secvenţe "clasice":

```
prog.Init;
prog.Run;
prog.Done.
```

Vă puteţi imagina un main mai simplu?

Fereastra este probabil cel mai versatil *view*. Descendenţii lui *TWindow* se bucură de calităţi speciale: pot fi "apucaţi" de un colţ şi deplasaţi pe ecran sau prinşi de colţul opus şi redimensionaţi. Ferestrele pot fi închise sau numai ascunse, cele numerotate (**n**) pot fi selectate cu combinaţia de taste **Alt-n** şi,evident, aruncă "umbre" pe suport. Desigur, o fereastră rămîne goală dacă nu este "umplută" de programator cu obiecte iar calităţile enumerate sînt de asemenea la discreţia lui, putîndu-şi personaliza ferestrele.

Amintit anterior, suportul - *TDesktop* - se bucură de două metode speciale, de mare efect, ce permit organizarea obiectelor subordonate redimensionabile în cascadă (suprapuse şi decalate precum un pachet de cărţi de joc) sau în domino (spaţiul suportului se împarte în părţi egale alocate fiecare cîte unui obiect). Windows a mai pierdut astfel două privilegii. Reprezentarea instantanee pe ecran şi absenţa pîlpîirilor determi-

nate de desenări succesive ne determină să săpăm mai adînc şi descoperim că TV întreţine cîte un buffer de memorie video-cache pentru fiecare grup, în care finalizează reprezentarea şi abia apoi "toarnă" imaginea pe ecran. Două metode, *Lock* şi *Unlock* sînt responsabile de ascunderea şi prezentarea cache-ului pe ecran. Să remarcăm de asemenea că memoria cache nu este numai dinamic alocată ci şi deplasată şi reorganizată pentru a evita fragmentarea memoriei libere rămase la dispoziţia programatorului.

Portrete

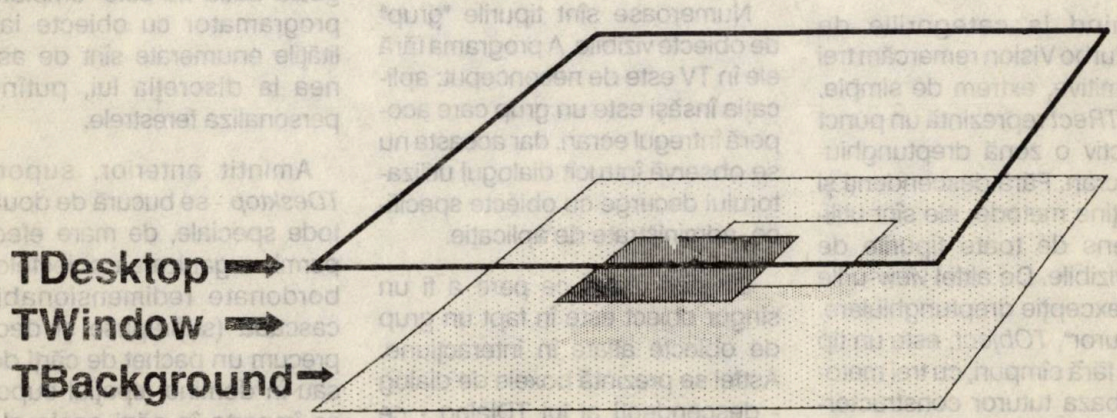
Dacă obiectele-grup se manifestă în background, consistenţa TV este conferită în cele din urmă de obiectele terminale - *view*-uri conţinute de grupuri şi avînd funcţii şi aspect specific. Un obiect vizibil terminal nu poate fi grup. El nu poate conţine alte obiecte vizibile şi încheie de aceea lanţurile de *view*-uri subordonate.

Butoanele au sarcina să asocieze evenimente de tastatură sau mouse cu o comandă. Ele generează o comandă specifică cînd sînt "apăsate" cu mouse-ul sau cu tasta activă corespunzătoare. Estetica lor este indiscutabilă: umbra dispare în momentul "apăsării".

Evidenţa standardizării unei interfeţe utilizator este dată de meniuri. Cu tipul *TApplication*, TV s-a aliniat la standardul Windows de meniuri pull-down; totuşi meniul linie - *TMenuBar* - şi meniul pop-up - *TMenuBox* - se pot înlănţui uşor pe oricîte nivele pentru a defini orice specie imaginabilă de meniu. Instanţierea unui meniu este elegantă: într-o singură instrucţiune se specifică lista înlanţuită de meniuri şi elemente de meniu, fiecare element indicîndu-l pe următorul. Evident meniurile sînt sensibile la mouse, dar implementează şi taste active (shortcuts); fiecărui element i se poate asocia un indice de context de help, care permite apariţia unei explicaţii pe linia status sau invocarea unei ferestre de help.

Între obiectele terminale, discrete dar foarte importante sînt barele de desfăşurare (*TScrollBar*) precum şi tipul de desfăşurător (*TScroller*). În filozofia TV în care prezentarea informaţiei pe ecran este optimizată prin posibilitatea de suprapunere a obiectelor, informaţia trebuie să poată fi desfăşurată sus-jos sau stînga-dreapta într-o fereastră de dimensiuni mici. Barele de desfăşurare pot furniza control pe verticală sau orizontală şi răspund atît la mouse cît şi la tastatură. Modul în care interacţionează prin mesaje barele de desfăşurare şi obiectele ce sînt desfăşurate vom avea ocazia să-l prezentăm mai jos. Funcţia foarte comună de inspectare şi selectare din liste este implementată prin *TListViewer*. Acest tip este asociat cu una sau două bare de desfăşurare iar selectarea unui articol se poate face atît cu mouse-ul cît şi cu săgeţile tastaturii. Utilizatorul poate redefini metodele *SelectItem* şi *FocusItem* pentru a îndeplini sarcini specifice la selectarea sau simpla focusare a unui articol; de asemenea poate interoga lista pentru a afla articolul curent. Tipul listei este la dispoziţia programatorului: listă înlănţuită de pointeri, vector de înregistrări, etc. *TListViewer* deţine o metodă specifică - *GetText* - prin care se defineşte modul de prezentare al articolului pe ecran. Un descendent - *TListBox* - implementează o boxă de vizualizare a listelor de string-uri, utilizată de foarte sofisticatele tipuri de obiecte de vizualizare şi selectare a numelor de fişiere şi directoare - *TFileDialog* şi *TChDirDialog*.

Liniile de introducere a datelor (*TInputLine*) sînt componente indispensabile pentru editarea unei linii de text sau în construcţia unei machete de culegere a datelor; suportă toate funcţiile standard de editare, inclusiv marcarea de blocuri cu ajutorul mouse-ului. Editarea poate avea loc într-o fereastră de dimensiuni mai mici decît lungimea liniei. Deseori le sînt asociate



Un grup poate fi conceput ca un sandwich de obiecte în care felia de deasupra este transparentă.

mecanisme de selecţie a unei valori anterior introduse - aşa numitele liste istorice (*THistory*, *THistoryWindow*, *THistoryViewer*).

Să privim un ultim "portret". Linia status - *TStatusLine* - este un obiect mai important decît s-ar putea crede: permite o afişare dinamică pe ecran, sensibilă la contextul aplicţiei şi implementează comenzi foarte importante (cum ar putea fi comanda de părăsire a aplicaţiei) ce trebuie să fie permanent accesibile. Cîmpul *Items* indică o listă înlănţuită de înregistrări conţinînd "etichete" asociate cu comenzi şi taste active (etichetele sînt, evident, sensibile la mouse). Un alt cîmp, *Defs* indică o listă de stringuri help ce pot fi asociate cu opţiuni de meniu prin intermediul indicilor de context help. TV trişează un pic cu liniile status: ele sînt active chiar şi în timpul unei execuţii modale, evenimentele şi comenzile generate de linia status fiind tratate ca şi cînd ar proveni de la obiectul modal.

Rîuri subterane de cod

Şi totuşi cum arată schema logică a unui program Turbo Vision? Unde începe execuţia şi unde se termină? În programarea clasică sîntem obişnuiţi să urmărim execuţia liniilor de program una după alta precum meandrele unui rîu de la izvor la vărsare.

În fapt avem de-a face cu o reţea de rîuri subterane de cod ce apar temporar la lumina zilei pe fundul unui aven pentru a dispare şi curge apoi pe căi neştiute. Programatorul nu are acces decît la cursul paralel, de suprafaţă, pe care el însuşi îl crează. Dar rîul propriu-zis curge în adînc.

De ce o asemenea structură? Pentru că o aplicaţie TV este prin excelenţă un program interactiv şi este greu de imaginat o altă arhitectură decît cea bazată pe tratarea evenimentelor. TV controlează în totalitate input-ul utilizatorului şi-l împachetează în înregistrări Pascal numite "evenimente" pe care le dispecerizează spre view-urile corespunzătoare. Programatorului nu-i mai rămîne decît să scrie cod care să ştie să trateze evenimentele intersante, fără a mai culege evenimente, fără a mai trata diferenţiat sursele de input, de la mouse sau tastatură.

Evenimentele pot fi concepute ca mici pachete de informaţie ce descriu semnale discrete la care aplicaţia trebuie să răspundă. Fiecare tastă apăsată, fiecare mişcare de mouse, o condiţie generată de altă componentă de program constituie un eveniment distinct. TV recunoaşte patru tipuri de evenimente: de mouse, tastatură, mesaje şi evenimente nule.

Mouse-ul poate genera următoarele evenimente: buton apăsat, buton eliberat, mouse mişcat, mouse "auto" (apăsarea continuă a unui buton), fiecare eveniment conţinînd informaţie relativă la poziţia mouse-ului pe ecran şi butonul În cauză. Evenimentele de tastatură sînt banale: apăsarea unei taste generează un eveniment ce conţine informaţie referitoare la poziţia tastei pe tastatură şi codul ASCII returnat.

Mesajele pot fi comenzi, emisiuni (*broadcasts*) şi mesaje utilizator; diferenţierea mesajelor ţine de modul în care sînt tratate. Cele mai multe evenimente sfîrşesc prin a fi transformate în comenzi. Selectarea unei opţiuni dintr-un meniu cu ajutorul unei taste active determină obiectul meniu să transforme evenimentul de tastatură într-un eveniment comandă. O comandă specifică este "interceptată" de obiectul "instruit" să o facă.

Comportamentul unui *view* este în totalitate determinat de o metodă numită *HandleEvent* ce primeşte ca argument evenimentul transmis obiectului. În această metodă programatorul defineşte modul în care obiectul trebuie să reacţioneze la primirea unor anumite comenzi sau evenimente de orice tip. Întrucît obiectul trebuie să aibă comportamentul specific al clasei căreia îi aparţine, este obligatorie moştenirea metodei *HandleEvent* a tipului de obiect părinte. Codul care trebuie să modifice comportamentul părintelui se inserează înainte de apelul *HandleEvent*, în timp ce codul care adaugă sarcini specifice va figura în program după apelul metodei. Există un set predefinit de comenzi TV care impun acţiuni standard: închide fereastră, selectează view, părăseşte program, etc. Programatorului îi rămîne însă un domeniu larg de definire a propriilor comenzi asociate unor acţiuni proprii aplicaţiei.

Pentru a înţelege şi celelalte tipuri de evenimente să urmărim cum sînt trimise evenimentele într-o aplicaţie. "Trimiterea" unui eveniment către un obiect înseamnă apelul metodei *HandleEvent* a acestuia, avînd ca argument evenimentul respectiv. Dacă obiectul receptor este la rîndul lui un grup el poate transmite evenimentul mai departe, invocînd metodele *HandleEvent* ale subobiectelor deţinute. Modul de dispecerizare diferenţiază evenimentele "poziţionale" de cele "focusate" şi de "emisiuni".

Evenimente poziţionale sînt (dacă programatorul nu decide altfel) numai evenimentele de mouse. Un eveniment poziţional este transmis obiectului modal (întotdeauna cel puţin unul într-o aplicaţie). Obiectul modal caută între *subview*-urile sale pe cel vizibil care conţine poziţia respectivă şi îi transmite evenimentul. Dacă *subview*-ul este la rîndul său un grup procesul continuă pînă cînd va fi găsit un obiect terminal sau poziţia respectivă nu mai este conţinută într-un *subview*. Ultimul obiect găsit va trata evenimentul.

Evenimentele "focusate" sînt în general evenimente de tastatură sau comenzi ce sînt transmise în jos, de-a lungul lanţului de obiecte selectate pînă la un obiect terminal. Să considerăm cazul unui editor de texte scris cu TV care este compus dintr-un obiect aplicaţie

care conţine pe lîngă meniu şi o linie status şi un suport, în care se pot deschide şi închide una sau mai multe ferestre de editare, fiecare conţinînd două bare de desfăşurare, un cadru şi un interior - editorul propriu-zis. Apăsarea unei taste generează un eveniment de tastatură care circulă de la obiectul modal curent - aplicaţia - În jos pe lanţul obiectelor selectate: suport (*TDesktop* este totdeauna obiectul selectat în *TApplication*), fereastra activă, editorul interior ferestrei (singurul obiect selectabil în fereastră). Evenimentul va sfirşi prin a fi tratat de ultimul obiect selectat - editorul interior ferestrei.

Şi totuşi cum reacţionează meniul aplicaţiei la taste active de meniu, sau cum sesizează bara de desfăşurare din fereastră tastele de desfăşurare (*PgUp*, *PgDn*), ambele fiind obiecte neselectabile în grupurile de care aparţin? Există două excepţii de la regula procesării evenimentelor focusate, bazate pe un flag de pre/postprocesare al *view*-ului în cauză, care determină trei faze ale parcursului:

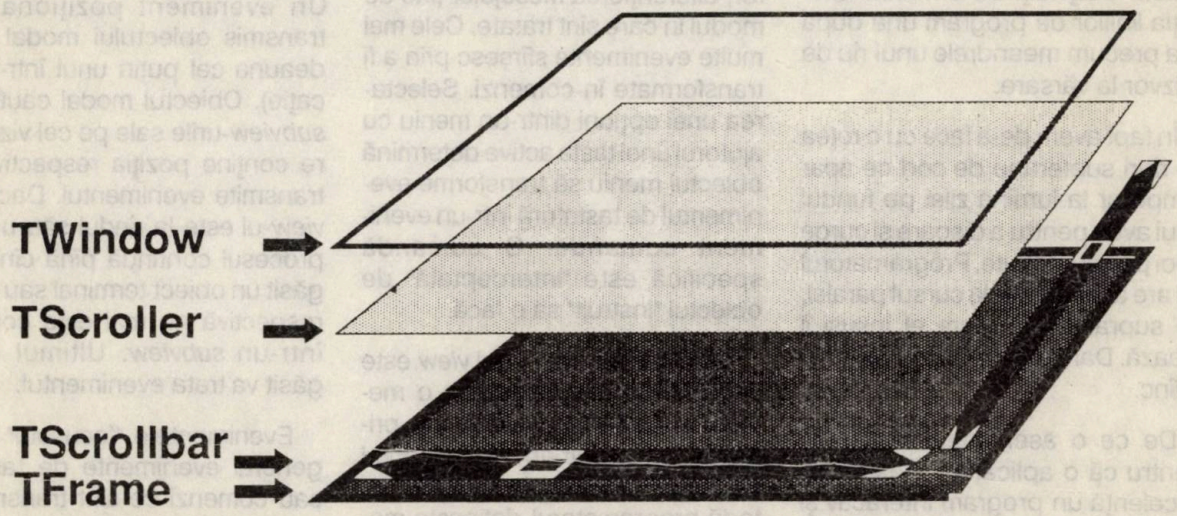
- - evenimentul este transmis tuturor *subview*-urilor din grup care au *flag*-ul de preprocesare setat (în ordinea inversă inserării lor); este cazul meniurilor;
- - dacă evenimentul nu a fost anulat în timpul preprocesării el parvine obiectului selectat din grup;
- - dacă evenimentul este în continuare valid el va fi transmis de obiectul grup tuturor *subview*-urilor avînd *flag*-ul de postprocesare setat (în ordinea inversă inserării lor în grup); astfel parvin evenimentele de tastatură barelor de desfăşurare.

S-ar părea că tehnica aceasta rezolvă problema concurenţei pentru un eveniment între obiectele conţinute pe acelaşi nivel al unui grup. Iată însă cazul unei boxe de dialog care conţine o linie de introducere date (*TInputLine*) şi un buton etichetat "OK". Pentru ca butonul să poată fi selectat cu combinaţia de taste **Alt-O** chiar şi atunci cînd linia de editare este obiectul curent selectat, el trebuie să primească primul evenimentele focusate, în faza de preprocesare. Întrucît butonul poate fi "apăsat" şi cu tasta O, acest eveniment trebuie tratat în faza de postprocesare, pentru a preveni imposibilitatea de a mai introduce caractere O în linia de editare. Cum recunoaşte butonul faza în care se află? Fiecare grup conţine un cîmp numit *Phase* avînd trei valori posibile, corespunzătoare fazei de procesare curente. Prin verificarea valorii cîmpului *Phase* al obiectului posesor, *TButton* poate afla în ce fază i s-a "pasat" evenimentul.

Un eveniment emisiune (*broadcast event*) nu-şi cunoaşte destinaţia şi este ca urmare trimis tuturor *subview*-urilor din obiectul modal curent. Emisiunea se propagă în jos pe arborele de *subview*-uri pînă la atingerea obiectelor terminale. Emisiunile au trebuit inventate pentru a facilita comunicaţia între obiecte, deziderat greu de realizat prin mijloace clasice în programarea OOP.

Un prim exemplu: barele de desfăşurare comunică listei noua poziţie a elementului focusat prin intermediul unui mesaj semnalizînd schimbarea, lansat spre grupul posesor, care-l va emite spre toate obiectele conţinute, dintre care numai lista va "şti" să-l interpreteze.

Cum se poate afla că un anumit obiect există şi nu mai este necesară instanţierea lui? Lansarea unui mesaj se face cu funcţia



Fereastra unui editor de texte grupează patru obiecte: editorul propriu-zis, totdeauna activ (selectat) în grupul ferestrei, cadrul exterior al ferestrei şi două bare de desfăşurare care se suprapun parţial cu cadrul.

Message după care controlul revine rutinei "emiţătoare" împreună cu adresa obiectului ce a interceptat mesajul sau un pointer nil in cazul în care obiectul căutat nu există. Semnalul universal al interceptării şi tratării unui eveniment este transformarea lui într-un eveniment nul. Dacă se doreşte ca un mesaj de genul "Stingerea!" să ajungă la toate obiectele de pe suport cerîndu-le "dispariţia", este suficient ca fiecare obiect să evite anularea evenimentului după tratare, permiţîndu-i să călătorească mai departe.

Programatorul poate defini cu uşurinţă noi tipuri de evenimente, cum ar fi cele asociate apariţiei unor caractere la portul serial. Implicit, TV le va trata ca emisiuni dar pot fi redefinite ca evenimente poziţionale sau focusate.

Puţini programatori începători în Turbo Vision rezistă tentaţiei de a specula *TProgram.Idle*, o spectaculoasă găselniţă a arhitecturii de tratare a evenimentelor. Această metodă primeşte controlul ori de cîte ori nu sînt evenimente disponibile şi permite instalarea unui proces paralel cu cel din *view*-ul curent. TV ne oferă două asemenea gadget-uri, un ceas şi un contor de memorie liberă. Deşi *Idle* nu oferă un multitasking veritabil, procesul din *view*-ul curent avînd prioritate absolută, totuşi În general calculatorul leneveşte în aşteptarea evenimentelor iar procesul secundar poate primi mai mult de 90% din timpul de execuţie

Ceea ce nu se vede

Dincolo de obiectele vizibile şi de traiectoriile evenimentelor, Turbo Vision păstrează un set de scule profesionale; stăpînirea lor nu este obligatorie pentru crearea unei aplicaţii TV, dar este o etapă inevitabilă pentru cei decişi să optimizeze codul pe linia OOP. Curentele, colecţiile şi resursele sînt toate tipuri derivate din *TObject*.

Colecţiile reprezintă alternativa TV la structurile de date clasice, de tipul vectorilor. *TCollection* este o listă înlănţuită de pointeri, realocabilă dinamic şi avînd un set de metode foarte rapide şi flexibile pentru operaţiuni standard de iterare, inserare, căutare. Un descendent - *TSortedCollection* - întreţine colecţii sortate după o cheie de index programabilă. Remarcabil este însă polimorfismul colecţiilor: elementele pot fi obiecte diferite şi de dimensiuni variabile; un strămoş abstract comun le conferă una sau mai multe metode virtuale ce permit exploatarea lor iterativă.

Încapsularea datelor cu codul în OOP a dus imediat la apariţia noţiunii de curent (*stream*) ca răspuns la întrebarea: de ce nu s-ar salva şi obiectele pe disc? În tehnicile tradiţionale trebuie scrise seturi distincte de funcţii pentru tratarea operaţiunilor I/O pentru tipuri diferite de date. Un curent TV permite salvarea şi citirea de obiecte derivate din *TObject*. Cheia recunoaşterii diverselor tipuri de obiecte dintr-un curent este înregistrarea prealabilă a tipului de obiect (tipurile standard TV sînt preînregistrate), utilizînd procedura *RegisterType*. *TStream* este un tip abstract de curent furnizînd un set de metode standard (*Get, Put, Reset, Error, CopyFrom, ReadStr, WriteStr*) şi şapte metode virtuale (*GetPos, GetSize, Seek, Truncate, Read, Write* şi *Flush*) definite în descendenţi specializaţi: *TDosStream* - curent de date stocate în fişiere DOS, *TBufStream* - versiune de *TDosStream* controlînd o memorie cache şi *TEMSStream* - curent de date stocate în memoria expandată.

O remarcabilă noutate în programare, preluată şi de Turbo Vision, o constituie noţiunea de resursă, un nume comun pentru elementele de program modificabile prin înlocuirea valorilor cîmpurilor din obiecte, incluzînd meniuri, boxe de dialog, liste de string-uri, etc. Resursele sînt în mod deliberat definite în afara codului într-o manieră relativ independentă de cod, astfel încît aplicaţiile pot deveni independente de limbă şi chiar de alfabet, cu minimum de modificări în cod. Resursele pot fi stocate în fişierul EXE, dar în afara codului executabil.

Programarea fără creare de resurse este de neimaginat în Turbo Pascal for Windows, dublat de un kit de editare interactivă a resurselor grafice. În absenţa unui asemenea kit pentru TV, se pot scrie deocamdată programe auxiliare de generare şi întreţinere a resurselor. Spre deosebire de alte porţiuni din Turbo Vision, nu este necesară redefinirea mecanismului resurselor, foarte robust şi flexibil.

Uşa din faţă

Turbo Vision se dovedeşte a fi destul de revoluţionar pentru a pune serioase probleme de adaptare programatorului clasic în Turbo Pascal. Neplăcuta surpriză de a nu putea depana pas cu pas o aplicaţie demonstrativă din kit-ul *TVDemos* poate descuraja un programator nerăbdător.

Cu regretul tipic românului sărac, trebuie să spunem începătorilor în OOP că referinţa originală Turbo Pascal 6.0 şi documentaţia Turbo Vision sînt cărţi absolut necesare. Nu este altă cale decît prin uşa din faţă. Ghidul TV este excelent scris şi rapid asimilabil, mai ales dacă urmaţi paşii propuşi de autori. Departe de a fi exhaustiv în ceea ce priveşte posibilităţile bibliotecii, el lasă loc la plăcute surprize ulterioare.

Cei experimentaţi în OOP pot găsi şi intrarea din spate, *help*-ul integrat compilatorului şi *demo*-urile fiind excelente. O referinţă a bibliotecii TV rămîne însă indispensabilă, aşa că, în limita spaţiului disponibil, vom încerca să vă oferim şi o astfel de referinţă.

(ing. Flaidăr Alin)

Pe urmele soluţiei

Programarea logică cu Turbo-Prolog, Partea a doua

După introducerea în principiile de bază ale programării logice din episodul anterior, de data aceasta vom examina tehnica de programare în Turbo-Prolog, prin intermediul unui exemplu.

O semnificaţie deosebită pentru Prolog o are operatorul relaţional **"not"**. Predicatul standard "not" este satisfăcut atunci cînd predicatul conţinut eşuează. De exemplu "not (5 > 7)" va furniza rezultatul "True", deoarece 5 nu este mai mare decît 7.

Totuşi "not" nu poate fi privit exact ca o negaţie matematică care inversează valoarea de adevăr. Prolog nu atribuie direct un "not" unui "goal" ci încearcă mai întîi să verifice predicatul conţinut. Dacă verificarea eşuează atunci predicatul "not" este satisfăcut.

Strategia decizională se bazează pe faptul că există ca dată o bază de cunoştinţe închisă, adică pe faptul că sistemului Prolog îi sînt puse la dispoziţie toate informaţiile necesare. Dacă nu se întîmplă aşa se poate întîmpla ca un goal să nu poată fi demonstrat şi deci să fie realizat non-goal-ul corespunzător.
Exemplu:

```
Domains
  persoana = symbol
predicates
  masculin (persoana)
  feminin (persoana)
clauses
  masculin (ion)
  feminin (X):- not (masculin (X)).
```

Dacă sistemului de mai sus i se pune întrebarea:

```
Goal: feminin (petru)
```

sistemul Prolog va răspunde cu YES, cu toate că "petru" nu este conţinut în baza de cunoştinţe. Trebuie remarcat şi faptul că în cazul lui "not" nu au voie să fie utilizate variabile care nu au fost legate încă de o valoare.

```
domains
  persoana = symbol
predicates
  sportiv (persoana)
  face_sport (persoana)
  fumeaza (persoana)
clauses
  face_sport (cristi).
  face_sport (gelu).
  fumeaza (gelu).
  sportiv (X):-not (fumeaza (X)),
    face_sport (X).
```

Dacă după acest exemplu se pune întrebarea "sportiv (X)" sistemul va răspunde pentru "not" sau "retractall" cu mesajul de eroare "704 Free variables are not allowed". Pentru a înlătura acest neajuns vor trebui operate modificările:

```
...
  sportiv (X):- face_sport (X),
    not (fumeaza (X)).
```

Prograamul modificat devine executabil deoarece variabila X este legată de o valoare înainte de apelul predicatului "not".

Pentru controlul execuţiei unui program Turbo-Prolog pune la dispoziţie predicatele standard **"cut"** şi **"fail"**.

Cu ajutorul lui cut (reprezentat printr-un semn de exclamare "!") în anumite condiţii poate fi limitată procedura de backtracking. Acest predicat este satisfăcut întotdeauna. Pentru aceasta trebuie luate în consideraţie două reguli:

1) atunci cînd operatorul cut este poziţionat la sfîrşitul unei clauze, atunci se va obţine o singură soluţie. Backtracking-ul, şi cu aceasta căutarea şi a unor alte soluţii, este întrerupt(ă).
2) dacă operatorul cut apare în interiorul unui predicat, atunci va fi execùtat backtracking-ul numai în spatele acestuia.

Următorul program exemplifică modul de utilizare al operatorului cut.

```
predicates
  numar (integer)
  combinatie (integer, integer, integer,
      integer)
clauses
  numar (1).
  numar (2).
  numar (3).
  numar (4).
  numar (5).
  combinatie (A,B,C,D):- numar (A),
    numar (B),
    A<B,
    !,
    numar (C),
    numar (D),
    <C, A<D,
    B<C, B<D,
    C<D.
```

Dacă pentru acest exemplu se interoghează:

```
Goal: "combinatie (A,B,C,D)"
```

se obţin următoarele soluţii:

```
A=1, B=2, C=3, D=4
A=1, B=2, C=3, D=5
A=1, B=2, C=4, D=5
3 Solutions
```

După o primă instanţiere a variabilelor A şi B, compilatorul mai caută combinaţii doar pentru variabilele C şi D. Dacă se înlătură operatorul cut, se obţin soluţiile:

```
A=1, B=2, C=3, D=4
A=1, B=2, C=3, D=5
A=1, B=2, C=4, D=5
A=1, B=3, C=4, D=5
A=2, B=3, C=4, D=5
5 Solutions
```

În acest caz sînt găsite două soluţii suplimentare deoarece backtracing-ul nu mai este limitat de operatorul cut.

Predicatul predefinit "fail" nu poate fi satisfăcut niciodată. Cu fail se forţează deci un backtracking. Acest lucru are însemnătate în cazul unui goal intern, cînd se doresc toate soluţiile posibile. Exemplu:

```
predicates
  numar (integer)
  combinatie (integer, integer, integer)
goal
  combinatie (X,Y,Z).
```

```
clauses
  numar (1).
  numar (2).
  numar (3).
  combinatie (X,Y,Z):-
    numar (X), numar (Y), numar (Z),
    X<>Y, Y<>Z, X<>Z,
    write (X,Y,Z), nl,
    fail.
```

Prin utilizarea lui fail se vor obţine toate combinaţiile posibile formate din cifre de la 1 la 3.

Puterea limbajului Prolog stă în prelucrarea nenumerică a datelor. Totuşi pot fi efectuate şi calcule matematice, asemănător modului în care se poate programa acest lucru cu limbajele convenţionale Pascal sau C. Pe lîngă operatorii aritmetici de bază (+, -, *, /, mod, div) şi operatorii aritmetici relaţionali (=, <, <=, >, >= şi <>) mai există o serie întreagă de funcţii şi predicate predefinite care pot fi utilizate pentru soluţionarea problemelor numerice.

Funcţii aritmetice:

sin (X), cos (X), tan (X), arctan (X),
log (X), ln (X), exp (X),
abs (X), sqrt (X), random (X), round (X),
bitand (X,Y,Z), bitor (X,Y,Z), bitxor (X,Y,Z),
bitnot (X, Y), bitleft (X,N,Z), bitright (X,N,Z)

Expresiile aritmetice, în Turbo-Prolog, se scriu cu ajutorul semnului de egalitate, de exemplu: "X = 25+7" va da soluţia "true" pentru X = 32.

Trebuie avută grijă deoarece, spre deosebire de Pascal sau C, în Prolog expresia poate fi interpretată ca o alocare de valoare. Dacă X are deja o valoare diferită de 32 predicatul care conţine această expresie eşuează. Dacă în acel moment variabila X este încă liberă, atunci va fi instanţiată cu valoarea 32. Această alocare de valori poate determina în anumite situaţii ca procesul de backtracking să mai facă un pas înapoi. Pentru a face, de exemplu, alocarea "X = Y - Z", variabila Y trebuie să fi fost deja instanţiată, altfel sistemul Prolog va afişa un mesaj de eroare specific.

În cazul unei expresii comparative, ca de exemplu "X <= Y - 2", ambele variabile trebuie să fi fost instanţiate.

Exemplul următor prezintă un scurt program aritmetic de soluţionare a unor ecuaţii de gradul doi:

```
predicates
  solutii (real, real, real)
clauses
  solutii (A,B,C):-
    B*B-4*A*C < 0,
    write ("Nu exista solutii"), nl.
  solutii (A,B,C):-
    B*B-4*A*C = 0,
    X = -B/(2*A),
    write ("Solutia: ",X), nl.
  solutii (A,B,C):-
    B*B-4*A*C > 0,
    X1 = (-B + sqrt (B*B-4*A*C))/2/A,
    X2 = (-B - sqrt (B*B-4*A*C))/2/A,
    write ("Solutii: ", X1," ", X2) nl.
```

Programul rezolvă ecuaţii de gradul doi de forma: "A * X * X + B * X + C = 0". Pentru aceasta el trebuie apelat cu goal-ul: "solutii (A,B,C)", unde A,B,C vor fi coeficienţii ecuaţiei. Cu "solutii (1,-2,1)" se vor obţine soluţiile ecuaţiei "X * X - 2 * X + 1 = 0". De predicatele de ieşire "write" şi "nl" ne vom ocupa mai tîrziu.

O listă este o înşiruire de obiecte ce constituie o unitate. Spre deosebire de cîmpurile din alte limbaje de programare, în Prolog listele sînt structuri dinamice. Aceasta înseamnă că în momentul iniţializării încă nu trebuie cunoscut numărul de elemente din care va fi formată lista.

Cu ajutorul listelor pot fi obţinute şi tipuri de date mai complexe cum sînt arborii şi grafurile. În Prolog listele sînt delimitate de paranteze drepte, elementele unei liste fiind separate prin virgulă. Exemplu: [0,1,2,3,4]. În Turbo-Prolog, elementele unei liste trebuie precedate de o instrucţiune DOMAINS care să le specifice tipul. Exemplu: "lista_de_intregi = integer*".

O listă constă din antet şi din corpul listei. Antetul este primul element din listă, iar corpul este format din restul listei. Exemplu:

Lista	Antet	Corp
[1,3,5,7]	1	[3,5,7]
[1]	1	[]
[[1,3,5],[7]]	[1,3,5]	[[7]]

O listă vidă este scrisă ca "[]". Antetul şi corpul ei nu sînt definite.

În Prolog antetul şi corpul unei liste pot fi separate cu ajutorul simbolului "|". Exemplu:

```
Daca:    [Antet|Corp] = [0,1,2,3,4,5]
atunci          Antet = 0
                Corp = [1,2,3,4,5]
```

Prin această separare a antetului de corpul unei liste pot fi accedate toate elementele listei. Următorul program exemplifică modul în care pot fi obţinute elementele unei liste:

```
domains
        lista_de_numere = integer *
predicates
        afisare_lista (lista_de_numere).
clauses
        afisare_lista ([]):- nl.
        afisare_lista ([Antet|Corp]):-
                write (Antet, " "),
                afisare_lista (corp).
```

Pentru execuţie programul trebuie apelat, de exemplu, cu goal-ul "afisare_lista ([6,4,78,36,23,9])."

Se vorbeşte despre recursivitate dacă o procedură se apelează pe ea însăşi. Acest lucru se întîmplă în Prolog atunci cînd antetul unei reguli apare şi în corpul regulii. Ca exemplu de apel recursiv vom da sursa unui numărător simplu:

```
predicates
  fara_sfirsit (integer)
goal
  fara_sfirsit (0).
clauses
  fara_sfirsit (X):- write (X,"),
    X1 = X+1,!,
    fara_sfirsit (X1).
```

Prin introducerea unei instrucţiuni de întrerupere, din bucla fără sfîrşit se obţine un numărător pînă la o valoare dată:

```
predicates
  fara_sfirsit (integer)
goal
  fara_sfirsit (0).
clauses
  fara_sfirsit (X):- write (X,"),
    X1 = X+1,
    X1 < 50,!,
    fara_sfirsit (X1).
```

Apelul va fi repetat pînă cînd X1 va lua valoare 50 şi condiţia "X1 < 50" nu va mai fi satisfăcută ceea ce va duce la întreruperea procesului recursiv.

Recursivitatea este folosită şi la definirea listelor. Calculul factorialilor este un exepmlu standard. Pro-

gramul următor calculează valoarea unui factorial cu ajutorul recursivităţii.

```
predicates
  factorial (integer, real)
  calculează_factorial
goal
  calculează_factorial.
clauses
  factorial (0,1).
  factorial (1,1).
  factorial (N, Rezultat):-
    N > 1,
    N1 = N-1,
    factorial (N1, RezultatN1),
    Rezultat = N * RezultatN1).
  calculeaza_factorial:-
    write ("Factorial de "),
    readint (X),
    factorial (X,Y),
    write ("Rezultat: ",Y), nl.
```

Turbo-Prolog dispune de o serie de predicate pentru citirea de date de la tastatură şi respectiv pentru afişarea datelor. În sistemul Prolog pot fi utilizate pentru intrări următoarele predicate:

- **readln (Rind)**, Citirea unei variabile de tipul şir sau de tipul simbol.
- **readint (Integer)**, Citirea unei variabile întregi.
- **readreal (Real)**, Citirea unei variabile de tipul real.
- **readchar (Char)**, Citirea unui singur caracter.
- **inkey (Char)**, Citeşte un caracter, dacă în fişierul de intrare există un caracter, altfel predicatul eşuează.
- **keypressed**, Cu acest predicat se poate stabili dacă o tastă a fost apăsată, fără să se citească caracterul tastat. keypressed reuşeşte atunci cînd o tastă oarecare este apăsată.

Pentru afişarea unor informaţii poate fi folosit predicatul **"write"** cu următoarea sintaxă de apel:

- write (Term 1, Term 2, Term 3, ...)

Numărul argumentelor predicatului "write" este oarecare. Cu el pot fi afişate constante sau variabile care au fost legate de o valoare.

Predicatul **"nl"** determină un avans de linie. În interiorul predicatului "write" pot fi utilizate următoarele comenzi cu backslah pentru controlul ieşirii:

- **/n** avans de linie
- **/t** tabulator
- **/z** afişează caracterul cu valoarea ASCII z
- // afişează un backslah

Cu ajutorul predicatului **"writef"** se pot afişa ieşiri formatate. Sintaxa predicatului "writef" este următoarea:

- writef (Format,Term 1,Term 2,...).

Formatul trebuie specificat în modul următor: **"%-m.pfeg"**, unde parametrii individuali au semnificaţiile:

- ieşire aliniată la stînga, dacă liniuţa lipseşte mesajul va fi aliniat la dreapta
- **m** lăţime minimă cîmp, nu este obligatoriu
- **p** precizia un număr în virgulă fixă sau lungimea maximă a unui şir
- **f** reprezentarea numerelor reale cu punct (standard)
- **e** reprezentarea numerelor reale cu virgulă
- **g** utilizează cel mai scurt format

Modul de utilizare al predicatelor de intrare/ieşire este exemplificat în programul următor:

```
predicates
  introduceredate (integer, real, string)
  afisaredate (integer, real, string)
  repeatuntilheypressed
  integerread (integer)
  realread (real)
  stringread (string)
goal
  introduceredate (X,Y,Z).
  afisaredate (X,Y,Z).
clauses
  repeatuntilheypressed:-
    write ("Pentru continuare apasa
      orice tasta..."),
    readchar (Ch), Ch=Ch, nl.
  integerread (X):-
    write :"Numar intreg:"), readint (X).
  realread (X):-
    write ("Numar real:"), readreal (X).
  stringread (X):-
    write ("Sir:"), readln (X).
  introduceredate (X,Y,Z):-
    integerread (X),
    realread (Y),
    stringread (Z),
    repeatuntilkeypressed.
  afisare_date (X,Y,Z):-
    writef ("aliniat stînga: %-10./n", X),
    writef ("aliniat dreapta: % 10./n", X),
    writef ("reprezentare cu punct:
      %-10.5f/n", Y),
    writef ("reprezentare cu virgula:
      %-19.5e/n", Y),
    writef ("aliniat dreapta: %-20.10/n", Z),
    writef ("aliniat stînga: %-20.10/n:, Z).
```

Pe lîngă operatorii logici **"şi"** şi **"sau"**

- A & B (conjuncţie)
- A V B (disjuncţie)

pot fi folosiţi şi alţi operatori ca de exemplu:

- A nor B (sau exclusiv)
- A <=> B (echivalenţă)
- A ==> B (implicaţie)

Valorile de adevărat sau fals (t=true, f=false) se dau cel mai adesea ca valori de adevăr.

Vă vom prezenta în continuare tabela de adevăr pentru operatorii logici şi combinaţiile lor:

A	t	t	f	f
B	t	f	t	f
A & B	t	f	f	f
A V B	t	t	t	f
A ==> B	t	f	t	t
A <== B	t	t	f	t
A <=> B	t	f	f	t
A	f	f	t	t
B	f	t	f	t
A nand B	f	t	t	t
A nor B	f	f	f	t
(A ==> B)	f	t	f	f
(A <== B)	f	f	t	f
A xor B	f	t	t	f
Contradiciţa	f	f	f	f

Operatorii anteriori pot fi reprezentanţi şi prin combinaţii de **"?"** şi **"v"**. Implicaţia "(A ==> B)" poate fi scrisă şi ca "(A v B). Operaţiile anterioare pot fi date în Prolog şi ca fapte dacă se iau valorile corespunzătoare din tabela de adevăr:

```
xor (0,0,0). /* sau exclusiv */
xor (0,1,1).
xor (1,0,1).
xor (1,1,0).
and (0,0,0). /* şi */
and (0,1,0).
and (1,0,0).
and (1,1,1).
```

Există o mulţime de legi ale algebrei booleene cu ajutorul cărora expresiile booleene pot fi reordonate pentru stabilirea valorii de adevăr a termenilor.

```
A & B <=> B & A  Legile comutativităţii
A V B <=> B V A
A & (B V C) <=> (A & B) V (A & C)
          Legile distributivităţii
A V (B & C) <=> (A V B) & (A V C)
(A & B) <=> (A) V (B)
          Legile lui de Morgan
(A V B) <=> (A) & (B)
A V A <=> A  Legile de idempotenţă
```

A & A <=> A
(A) <=> A Dubla negaţie

Aceste legi pot fi uşor verificate cu ajutorul tabelelor de adevăr. Implicaţia "A1,A2,A3,...==> B", corespunde clauzei Prolog "B:-A1,A2,A3,...", denumită şi clauză Horn. Scopul programării logice este următorul: obţinerea tuturor concluziilor logice sub formă de clauze Horn.

Pe lîngă formulele logicii declarative, Prolog lucrează şi cu variabile. Acest domeniu este desemnat ca logica predicatelor. Formele declarative sînt date cu aşa-numiţii "cuantificatori universali" şi "cuantificatori de existenţă". (Cuantificator universal:...Este valabil pentru toate; Cuantificator de existenţă: Există o /un...pentru care este valabil). Cuantificatorii sînt reprezentaţi în Prolog cu ajutorul variabilelor.

Să luăm în considerare declaraţiile:

- - toţi oamenii sînt muritori
- - există un om cu numele Cezar
- - Cezar este muritor.

Programul Prolog corespunzător va arăta astfel:

```
predicates
   este_om (symbol)
   este_muritor (symbol)
clauses
   este_om (cezar).
   este_muritor (X):-
      este_om (X).
```

Goal-ul "este_muritor (X)" va furniza "X=cezar".

Ca urmare Prolog poate fi utilizat pentru rezolvarea problemelor logice. Să luăm un mic exemplu:

Patru fraţi Ion, Petru, Gabriel şi Alexandru se joacă în curte cu mingea şi sparg un geam. Mama îi întreabă cum s-a întîmplat şi primeşte următoarele răspunsuri:
Ion spune că Petru şi Gabriel mint.
Petru spune că Ion şi Gabriel mint.
Gabriel spune că Ion, Petru şi Alexandru mint.
Alexandru spune că Ion, Petru şi Gabriel mint.
Pe cine poate să creadă mama?

La această întrebare se poate răspunde cu ajutorul sistemului Prolog. Din răspunsul lui Ion se obţine următorul predicat Prolog: "doi_mint (Ion, Petru, Gabriel)". De asemenea din al doilea enunţ se obţine: "doi_mint (Petru, Ion, Gabriel)". Pentru predicatul "doi_mint" există următoarele patru posibilităţi:

1) Ion spune adevărul, caz în care Petru şi Gabriel mint,
2) Ion minte, caz în care
   a) fie Petru şi Gabriel spun amîndoi adevărul,
   b) fie Petru minte şi Gabriel spune adevărul
   c) fie Petru spune adevărul şi Gabriel minte.

Aceasta implică următoarele fapte pentru predicat:

```
doi_mint (cinstit, mincinos, mincinos)
doi_mint (mincinos, cinstit, cinstit)
doi_mint (mincinos, mincinos, cinstit)
doi_mint (mincinos, cinstit, mincinos)
```

Unde "mincinos" desemnează o persoană care minte, iar "cinstit" desemnează o persoană care spune adevărul. Pentru următoarele două enunţuri poate fi scris predicatul "trei_mint". Enunţurile vor fi modelate în modul următor:

```
trei_mint (Gabriel, Ion, Petru, Alexandru)
trei_mint (Alexandru, Ion, Petru, Gabriel)
```

În mod analog ca şi în cazul predicatului "doi_mint", în cazul acestui predicat sînt posibile următoarele opt fapte:

```
trei_mint (mincinos, mincinos, mincinos,
    cinstit)
trei_mint (mincinos, mincinos, cinstit,
    mincinos)
trei_mint (mincinos, cinstit, mincinos,
    mincinos)
trei_mint (mincinos, mincinos, cinstit,
    cinstit)
trei_mint (mincinos, cinstit, mincinos,
    cinstit)
trei_mint (mincinos, cinstit, cinstit,
    mincinos)
trei_mint (mincinos, cinstit, cinstit, cinstit)
trei_mint (cinstit, mincinos, mincinos,
    mincinos)
```

Cine minte şi cine spune adevărul va putea fi stabilit cu ajutorul următoarei reguli:

```
cine_minte (Ion, Petru, Alexandru):-
doi_mint (Ion, Petru, Gabriel),
doi_mint (Petru, Ion, Gabriel),
trei_mint (Gabriel, Ion, Petru, Alexandru),
trei_mint (Alexandru, Ion, Petru, Gabriel).
```

Astfel se obţine următorul program Prolog:

```
predicates
   doi_mint (symbol, symbol, symbol)
   trei_mint (symbol, symbol, symbol,
       symbol)
   cine_minte (symbol, symbol, symbol,
       symbol)
clauses
   doi_mint (mincinos, mincinos, cinstit).
   doi_mint (mincinos, cinstit, mincinos).
   doi_mint (mincinos, cinstit, cinstit).
   doi_mint (cinstit, mincinos, mincinos).
   trei_mint (mincinos, mincinos, mincinos
       cinstit).
   trei_mint (mincinos, mincinos, cinstit,
       mincinos).
   trei_mint (mincinos, cinstit, mincinos,
       mincinos).
   trei_mint (mincinos, mincinos, cinstit,
       cinstit).
   trei_mint (mincinos, cinstit, mincinos,
       cinstit).
   trei_mint (mincinos, cinstit, cinstit,
       mincinos).
   trei_mint (mincinos, cinstit, cinstit, cinstit).
   trei_mint (cinstit, mincinos, mincinos,
       mincinos).
   cine_minte (Ion, Petru, Gabriel,
      Alexandru):-
       doi_mint (Ion, Petru, Gabriel),
       doi_mint (Petru, Ion, Gabriel),
       trei_mint (Gabriel, Ion, Petru,
          Alexandru),
       trei_mint (Alexandru, Ion, Petru,
          Gabriel).
```

Pentru a afla cine minte şi cine spune adevărul programul trebuie lansat în execuţie astfel:

```
Goal: cine_minte (Ion, Petru, Gabriel,
    Alexandru)
```

şi se va obţine:

```
Ion = mincinos, Petru = mincinos,
Gabriel = cinstit, Alexandru = mincinos
1 Solution
```

În episodul următor vom descrie modul în care se poate programa în Turbo-Prolog apropiat de codul maşinii pe care se lucrează. Turbo-Prolog dispune de predicate speciale cu ajutorul cărora utilizatorul poate programa la nivelul sistemului de operare.

De asemenea ne vom ocupa de modul în care pot fi adăugate unui program Prolog proceduri scrise în alte limbaje de programare.

(R.M.)

Clase, obiecte şi moşteniri

Nici o filozofie de programare nu a stîrnit atîta interes în ultimul timp ca programarea orientată obiect. După cursul introductiv pentru limbajul Pascal, vă prezentăm acum continuarea pentru cei interesaţi de C++.

Transpunerea acestei noi filozofii în soft nu se opreşte în faţa nici unui limbaj de programare. Ce se află îndărătul conceptiei orientate obiect? Doar şi pînă acum, oricînd, datele de prelucrat puteau fi abstractizate într-atît încît să poată fi gestionate eficient! Şi totuşi, cine descoperă noile extensii de limbaj, e nevoit să-şi schimbe părerile.

Cine vorbeşte de C++, se referă la un C orientat pe clase, deoarece, în acest limbaj, obiectele sînt denumite clase. Analogia cu biologia vine de la sine: şi aici se vorbeşte de clase, subclase, specii, etc. Această analogie vine în ajutorul începătorilor, permiţînd o acomodare mai rapidă cu noţiunile neobişnuite de care, inevitabil, se lovesc.

Clasele servesc perfecţionării modelelor. Modelele biologice existente sînt supuse - chiar şi după ani şi ani de de cercetare - unor modificări sau unor extensii.

În limbajele de programare uzuale, pentru date cu caracteristici identice, dar de tipuri diferite, trebuiesc create funcţii diferite. Adunarea numerelor întregi, reale sau complexe se deosebeşte prin modul de predare al tipului de date. Conform ANSI C, parametrii de apel trebuie să corespundă celor din funcţia apelată. Funcţiile primesc nume diferite, chiar dacă principial fac acelaşi lucru, doar cu date de alt tip. În C nu există altă posibilitate, decît a produce trei funcţii, cu nume diferite.

Cauza pentru întărirea controlului asupra tipului datelor este posibila transmitere - generatoare de erori - a unor date de alt tip decît cele aşteptate către o funcţie. Este exact ce se evită prin încapsularea datelor şi a codului aferent lor. Funcţii-element care fac acelaşi lucru (de exemplu, adună) pot avea acelaşi nume. În C++ este chiar indicat să se folosească această extensie - denumită, uzual "supraîncărcare" (overload). Este, în principiu, la latitudinea programatorului, în ce măsură foloseşte avantajele programării orientate obiect. Este important să se înţeleagă şi să se folosească aceste avantaje. Efectul va fi un control mai bun asupra structurii programelor şi modularitate. În C++ pot fi generate tipuri de date noi, cu operatori speciali, care se potrivesc numai lor.

Caracteristicile programării orientate obiect sînt sintetizate în noţiunile de încapsulare, ereditate şi polimorfie. Ce înseamnă acestea, mai în amănunt, se poate vedea din fig. 2. Unii experţi sînt de părere că, pentru a învăţa C++, trebuie uitat tot ce s-a învăţat mai înainte (ceea ce ei, cu siguranţă, nu au făcut...). Parţial au dreptate, parţial nu. Apropierea de C++ se face cel mai bine adăugînd la cunoştinţele de bază despre C extensiile de limbaj orientate obiect, la fel cum s-ar face acest lucru în Pascal.

După aceste consideraţii preliminare, prima încercare OOP. Este vorba de adunarea descrisă, în cazul unor date de tipuri diferite.

Programul din listingul ADD.CPP produce următoarele mesaje pe ecran:

Prima sumă este 5.000
A doua sumă este 7.264
A treia sumă este 2.001.

La prima adunare, funcţiei ADD îi sînt date două valori întregi. Înaintea celei de-a doua adunări, sînt definite două numere în virgulă mobilă. Această adunare poate fi executată numai cu cea de-a doua funcţie ADD. La versiuni mai vechi de C++, funcţia supraîncărcată trebuia declarată cu cuvîntul cheie prefixat "overload" Acesta a fost declarat depăşit între timp.

Cele două funcţii ADD permit revelarea unei alte caracteristici importante a C++. Parametrilor funcţiei li se pot ataşa valori iniţiale. Dacă înaintea celei de-a treia adunări ai doilea termen al adunării n-ar fi fost iniţializat cu zero, rezultatul ar fi fost greşit. Interesant este că funcţii supraîncărcate pot fi folosite şi independent de clase speciale. Dar să nu anticipăm.

Cum se definesc clasele? C++ dispune pentru aceasta de cuvîntul cheie "class". Dar şi vechile cuvinte cheie "struct" respectiv "union" pot fi folosite în acest scop. După unul din aceste trei cuvinte cheie urmează denumirea clasei ce se defineşte. Exemplu:

```
class student {
    char facultate [30];
    int semestru;
};
```

Pentru clasa "student" au fost stabilite atributele (data members) "facultate" şi "semestru". Majoritatea claselor sînt derivate pornind de la clase de bază. Să presupu-

```
1  | // exemple de functii supraincarcate
2  | // adunare intregi sau . in virgula mobila
3  |
4  | #include <stdio.h>
5  |
6  | int add(int add1 = 0, int add2 = 0)
7  | {
8  |  return (add1 + add2);
9  | }
10 |
11 | //overload
12 | float add (float add1 = 0, float add2 = 0)
13 | {
14 |  return (add1 + add2);
15 | }
16 |
17 | int main()
18 | {
19 |  float x, y, z;
20 |
21 |  z = add (2, 3);
22 |  printf ("\n Prima suma este: %6.3f",z);
23 |  x = 2.001; y = 5.263; z = add (x,y);
24 |  printf ("\n A doua suma este: %6.3f",z);
25 |  z = add(x);
26 |  printf ("\n A treia suma este: %6.3f",z);
27 | }
```

ADD.CPP

```
// Listing exemplu nr.1
// un obiect cu functii-element

#include <conio.h>
#include <stdio.h>

class Semn  // definitia clasei Semn
{
 int X, Y;          coordonate
 char valoare;  // val. alfanumerica
 char attr;         // atribut ecran

 public:
  Semn (int InitX, int InitY)
  {
    X = InitX;
    Y = InitY;
    valoare = ' ';
  }

  // Functii-element publice

  int GetX ()  { return X; }
  int GetY ()  { return Y; }
 void Position (int PosX, int PosY)
  {
    X = PosX; Y = PosY;
  }
  void Citire ()
  {
   gettext (X, Y, X, Y, &valoare);
  }
  void Scriere (char Sign)
  {
   valoare = Sign;
   puttext (X, Y, X, Y, &valoare);
  }
};

int main ()
{
 int X1, Y1;

 clrscr ();
 X1 = 4; Y1 = 10;
 Semn ASemn (X1, Y1);
 ASemn.Citire ();
 ASemn.Scriere ('A');
 printf ("\n Apasati o tasta ... ");
 getch ();
 return 0;
}
```

EX1.CPP

```
// Listing exemplu nr. 2
// alte citeva functii-element

#include <conio.h>
#include <stdio.h>

class Semn  // definitia clasei Semn
{
 int X, Y;   // coordonate
 char valoare; // val. alfanumerica
 char attr;  // atribut ecran

 public:
  Semn (int InitX, int InitY)
  {
    X = InitX;
    Y = InitY;
    valoare = ' ';
  }

  // Functii-element publice

  int GetX ()  { return X; }
  int GetY () { return Y; }
 void Position (int PosX, int PosY)
  {
    X = PosX; Y = PosY;
  }
  void Citire ()
  {
   gettext (X, Y, X, Y, &valoare);
  }
  void Scriere (char Sign)
  {
   valoare = Sign;
   puttext (X, Y, X, Y, &valoare);
  }
  void Indicare ()
  {
   puttext (X, Y, X, Y, &valoare);
  }
  void Stergere ()
  {
   char memo;

   memo = valoare; valoare = ' ';
   Indicare ();
   valoare = memo;
  }
 void Miscare (int NewX, int NewY)
  {
   Stergere();
   Position (NewX, NewY);
   Indicare ();
  }
};

int main ()
{
 int X1, Y1;

 clrscr ();
 X1 = 4; Y1 = 10;
 Semn ASemn (X1, Y1);
 ASemn.Citire ();
 ASemn.Scriere ('A');

 printf ("\n Apasati o tasta ... ");
 getch ();
 ASemn.Miscare (X1+4, Y1-4);
 getch();
 ASemn.Scriere ('B');
 getch();
 ASemn.Stergere();
 getch(); return 0;
}
```

EX2.CPP

nem că există o clasă de bază "persoană" cu următoarele atribute:

```
class persoana {
    char nume [30];
    char prenume [20];
    int virsta;
 }
```

Atunci ar fi posibilă derivarea clasei "student" din clasa "persoana". Pentru aceasta, ar trebui modificată definiţia clasei "student":

```
class student:
   persoana {
     char facultate [30];
     int semestru;
  };
```

Clasa "student" moşteneşte toate caracteristicile (atributele) clasei "persoana", reprezintă însă o clasă derivată.

Să precizăm că modul de acces la caracteristicile individuale poate fi definit altfel în cele două clase.

După aceste consideraţii a devenit limpede cum se defineşte o clasă simplă. Forma generală este:

```
<classkey>
<classname>
< : baselist > {
  <member list>
}
```

Dar ce este o clasă fără metode care să facă posibilă preluarea şi modificarea atributelor sale? Eficienţa OOP se obţine, în principal, tocmai prin moştenirea de către clasele derivate a metodelor proprii clasei "mamă".

Exemplul conţine definiţia clasei "Semn". Clasa dispune de 4 caracteristici: coordonatele x şi y pe ecran, un caracter şi un atribut al acestuia. Valorile sînt păstrate în memoria de împrospătare a imaginii.

Urmează declararea funcţiilor-element. Atunci cînd descrierea funcţiei -element urmează imediat definiţiei ei, se vorbeşte de funcţii incorporate (inline). "GetX ()" este descrisă complet în interiorul clasei "Semn", între delimitatorii ă şi î, prin intermediul lui return X; funcţii de acest fel, pot fi precedate de cuvîntul cheie "inline":

```
inline int GetX ()
{ return X;
}
```

Dacă funcţia este definită în afara clasei, atunci în descrierea defi-

niţiei trebuie să se facă referire la clasa căreia funcţia îi aparţine:

```
int Semn:: GetX ()
{ return X;
};
```

În listingul reprodus există metodele necesare preluării coordonatelor precum şi pentru citirea şi scrierea caracterelor în mod text. La citire, se preia şi atributul ecran existent.

Se poate urmări şi încapsularea datelor împreună cu codul ce permite manipularea lor.

În funcţiile "Citire ()" şi "Scriere ()" se foloseşte operatorul "&" pentru referinţe. Acesta se foloseşte asemănător cu un pointer. La transmiterea argumentelor nu se transmite valoarea lor, ci adresa de referinţă la valoare.

Referinţele sînt pointeri spre obiecte, putînd fi utilizate (oarecum) în locul acestora.

Definiţia unei clase reprezintă declararea unui tip de date specific utilizatorului, iar obiectul este o variabilă de acest tip. În exemplu, se declară variabila "ASemn" de tip "Semn". Acest obiect este static, deoarece spaţiul de care are nevoie i se alocă încă din timpul compilării. Obiectelor dinamice li se alocă memoria necesară pe stivă, în momentul execuţiei - cînd îşi termină existenţa, memoria de care au avut nevoie este eliberată. Observaţii mai în detaliu referitoare la obiecte dinamice vor fi prezentate într-o parte separată a cursului.

În program, se declară obiectul "ASemn" avînd coordonatele x=4 şi y=10. Apoi, se citeşte caracterul aflat în acel loc, împreună cu atributul lui. Apoi, caracteristicii "valoare" a obiectului i se dă valoarea "A", afişarea făcîndu-se păstrînd vechiul atribut. Pentru afişare, se foloseşte metoda "Scriere" aplicată obiectului "ASemn".

· În exemplul 2, clasei "Semn" i-au fost adăugate alte cîteva funcţii-element. Scopul a fost mişcarea obiectului în alt loc pe ecran. Pentru aceasta a trebuit să fie posibilă ştergerea din poziţia veche, transmiterea noilor coordonate şi afişarea obiectului în noua poziţie. Metoda "Indicare ()" se deosebeşte de "Scriere (char Sign)" prin faptul că nu se transmite o valoare nouă. În acest exemplu, "Ştergere ()" semnifică afişarea unui spaţiu în punctul de coordonate x, y.

Obiectului "ASemn" i se predă mesajul "Mişcare (x1+4, y1-4)". Urmează şi alte mesaje, simplu de înţeles. Mesajele denumite în acest fel trebuie să ajute utilizatorul să înţeleagă sensul programului. Prin OOP, "transparenţa" aplicaţiei poate fi îmbunătăţită.

Mai interesant devine Exemplul 3. Nici un obiect nu există numai pentru sine. Exemplul vrea să ilustreze interacţiunea dintre obiecte pe un caz simplu.

Obiectul "BSemn" este copiat pe tot ecranul, astfel încît fondul apare ca o plasă. Obiectul "ASemn" ur-

```
1   | // Listing exemplu nr. 3
2   | // extind. caract. obiectului
3   | // ptr. prez. interact. intre obiecte
4   |
5   | #include <conio.h>
6   | #include <stdio.h>
7   |
8   | class Semn  // definitia clasei Semn
9   | {
10  |  int X, Y;          // coordonate
11  |  char valoare;  // val. alfanumerica
12  |  char attr;         // atribut ecran
13  |  char valoare_veche;
14  |  char attr_vechi;
15  |
16  |  public:
17  |   Semn (int InitX, int InitY)
18  |   {
19  |    X = InitX;
20  |    Y = InitY;
21  |   }
22  |
23  |   // Functii-element publice
24  |
25  |   int GetX () { return X; }
26  |   int GetY () { return Y; }
27  |  void Position (int PosX, int PosY)
28  |   {
29  |    X = PosX; Y = PosY;
30  |   }
31  |   void Citire ()
32  |   {
33  | gettext (X, Y, X, Y, &valoare_veche);
34  |   }
35  |   void Scriere (char Sign)
36  |   {
37  |    Citire ();
38  |    valoare = Sign;
39  |    puttext (X, Y, X, Y, &valoare);
40  |   }
41  |   void Modificare (char Sign)
42  |   {
43  |    valoare = Sign;
44  |    puttext (X, Y, X, Y, &valoare);
45  |   }
46  |   void Indicare ()
47  |   {
48  |    puttext (X, Y, X, Y, &valoare);
49  |   }
50  |   void Stergere ()
51  |   {
52  |    valoare = valoare_veche;
53  |    attr = attr_vechi;
54  |    Indicare ();
55  |   }
56  |  void Miscare (int NewX, int NewY)
57  |   {
58  |    char memo_valoare;
59  |    char memo_attr;
60  |
61  |    memo_valoare = valoare;
        |      valoare = valoare_veche;
62  |    memo_attr = attr;
        |     attr = attr_vechi;
63  |    Indicare ();
64  |    valoare = memo_valoare;
65  |    attr = memo_attr;
66  |    Position (NewX, NewY);
67  |    Citire ();
68  |    Indicare ();
69  |   }
70  |  void Copiere (int NewX, int NewY)
71  |   {
72  |    Position (NewX, NewY);
73  |    Indicare ();
74  |   }
75  |   void Seteaza_attr (char attr_nou)
76  |   {
77  |    attr = attr_nou;
78  |    Indicare ();
79  |   }
80  | };
81  |
82  | int main ()
83  | {
84  |  int X1, Y1, s, z;
85  |
86  |  X1 = 1; Y1 = 1;
87  |  Semn BSemn (X1, Y1);
88  |  BSemn.Scriere ('+');
89  |  BSemn.Seteaza_attr (BLUE);
90  |  for (s = 0; s < 25; s++)
91  |   for (z = 0; z < 80; z++)
92  |    BSemn.Copiere (X1+z, Y1+s);
93  |  X1 = 4; Y1 = 10;
94  |  Semn ASemn (X1, Y1);
95  |  ASemn.Scriere ('A');
96  |  ASemn.Seteaza_attr (WHITE);
97  |  printf ("\n Apasati o tasta ... ");
98  |  getch ();
99  |  ASemn.Miscare (X1 + 4, Y1 - 4);
100 |  getch ();
101 |  ASemn.Modificare ('B');
102 |  getch ();
103 |  ASemn.Stergere ();
104 |  getch (); clrscr ();
105 |  return 0;
106 | }
```

EX3.CPP

```
1   | // Listing exemplu nr. 4
2   | // def. a doua obiecte diferite
3   | // unul fiind derivat din celalalt
4   | // Semn --> Semn2
5   |
6   | #include <conio.h>
7   | #include <stdio.h>
8   |
9   | class Semn   // definitia clasei Semn
10  | {
11  |  public:
12  |   int X, Y;       // coordonate
13  |   char valoare; // val. alfanumerica
14  |   char attr;      // atribut ecran
15  |
16  |   Semn (int InitX, int InitY)
17  |   {
18  |     X = InitX;
19  |     Y = InitY;
20  |   }
21  |
22  |   // Functii-element publice
23  |
24  |   int GetX () { return X; }
25  |   int GetY () { return Y; }
26  |   void Position (int PosX, int PosY)
27  |   {
28  |     X = PosX; Y = PosY;
29  |   }
30  |   void Citire ()
31  |   {
32  |    gettext (X, Y, X, Y, &valoare);
33  |   }
34  |   void Scriere (char Sign)
35  |   {
36  |    Citire ();
37  |    valoare = Sign;
38  |    puttext (X, Y, X, Y, &valoare);
39  |   }
40  |   void Modificare (char Sign)
41  |   {
42  |    valoare = Sign;
43  |    puttext (X, Y, X, Y, &valoare);
44  |   }
45  |   void Indicare ()
46  |   {
47  |    puttext (X, Y, X, Y, &valoare);
48  |   }
49  |   void Stergere ()
50  |   {
51  |    valoare = ' ';
52  |    Indicare ();
53  |   }
54  |  void Miscare (int NewX, int NewY)
55  |   {
56  |    char memo_valoare;
57  |
58  |    memo_valoare = valoare;
59  |    Stergere ();
60  |    valoare = memo_valoare;
61  |    Position (NewX, NewY);
62  |    Indicare ();
63  |   }
64  |  void Copiere (int NewX, int NewY)
65  |   {
66  |    Position (NewX, NewY);
67  |    Indicare ();
68  |   }
69  |   void Seteaza_attr (char attr_nou)
70  |   {
71  |    attr = attr_nou;
72  |    Indicare ();
73  |   }
74  | };
75  |
76  | // definitia clasei Semn2
77  |
78  | class Semn2 : public Semn
79  | {
80  |  char valoare_veche;
81  |  char attr_vechi;
82  |
83  |  public:
84  | Semn2 (int InitX, int InitY) :
          Semn (InitX, InitY)
85  |  {
86  |   valoare_veche = 0x0;
87  |   attr_vechi = 0x0;
88  |  }
89  |
90  |  // Functii-element publice
91  |
92  |   void Citire ()
93  |   {
94  | gettext (X, Y, X, Y, &valoare_veche);
95  |   }
96  |   void Scriere (char Sign)
97  |   {
98  |    Citire ();
99  |    valoare = Sign;
100 |    puttext (X, Y, X, Y, &valoare);
101 |   }
102 |   void Modificare (char Sign)
103 |   {
104 |    valoare = Sign;
105 |    puttext (X, Y, X, Y, &valoare);
106 |   }
107 |   void Indicare ()
108 |   {
109 |    puttext (X, Y, X, Y, &valoare);
110 |   }
111 |   void Stergere ()
112 |   {
113 |    valoare = valoare_veche;
114 |    attr = attr_vechi;
115 |    Indicare ();
116 |   }
117 |  void Miscare (int NewX, int NewY)
118 |   {
119 |    char memo_valoare;
120 |    char memo_attr;
121 |
122 |    memo_valoare = valoare;
          valoare = valoare_veche;
123 |    memo_attr = attr;
          attr = attr_vechi;
124 |    Indicare ();
125 |    valoare = memo_valoare;
126 |    attr = memo_attr;
127 |    Position (NewX, NewY);
128 |    Citire ();
129 |    Indicare ();
130 |   }
131 |  void Copiere (int NewX, int NewY)
132 |   {
133 |    Position (NewX, NewY);
134 |    Indicare ();
135 |   }
136 |  void Seteaza_attr (char attr_nou)
137 |   {
138 |    attr = attr_nou;
139 |    Indicare ();
140 |   }
141 | };
142 |
143 | int main ()
144 | {
145 |  int X1, Y1, s, z;
146 |
147 |  X1 = 1; Y1 = 1;
148 |  Semn BSemn (X1, Y1);
149 |  BSemn.Scriere ('+');
150 |  BSemn.Seteaza_attr (BLUE);
151 |  for (s = 0; s < 25; s++)
152 |   for (z = 0; z < 80; z++)
153 |    BSemn.Copiere (X1+z, Y1+s);
154 |  X1 = 4; Y1 = 10;
155 |  Semn2 ASemn (X1, Y1);
156 |  ASemn.Scriere ('A');
157 |  ASemn.Seteaza_attr (WHITE);
158 |  printf ("\n Apasati o tasta ... ");
159 |  getch ();
160 |  ASemn.Miscare (X1 + 4, Y1 - 4);
161 |  getch ();
162 |  ASemn.Modificare ('B');
163 |  getch ();
164 |  ASemn.Stergere ();
165 |  getch (); clrscr ();
166 |  return 0;
167 |}
```

EX4.CPP

mează să citească şi să memoreze informaţia ecran anterioară, pentru ca la ştergerea sau mişcarea sa să se vadă din nou fundalul (plasa să nu se găurească). Pentru aceasta, sînt introduse pur şi simplu două noi atribute pentru obiectul "Semn". Mai trebuie în plus modificate funcţiile-element "Scriere ()" şi "Ştergere ()". Înainte de scriere, trebuiesc reţinute valoarea veche şi atributul existent. Ştergerea în cadrul acestei clase modificate înseamnă acum afişarea valorii originale. Dacă obiectul "ASemn" urmează să primească o altă valoare în aceeaşi poziţie, acest lucru nu are voie să se întîmple prin metoda "Scriere ()", deoarece în acest fel s-ar strica valoarea memorată iniţial în "valoare_veche". De aceea s-a introdus o altă metodă, care permite modificarea valorii existente în aceeaşi

```
1   | // Listing exemplu nr. 5
2   | // functii-element mostenite
3   |
4   | #include <conio.h>
5   | #include <stdio.h>
6   |
7   | class Semn  // definitia clasei Semn
8   | {
9   |   int X, Y;        // coordonate
10  |  public:
11  |  char valoare;  // val. alfanumerica
12  |   char attr;        // atribut ecran
13  |
14  |   Semn (int InitX, int InitY)
15  |   {
16  |     X = InitX;
17  |     Y = InitY;
18  |   }
19  |
20  |   // Functii-element publice
21  |
22  |   int Semn::GetX () { return X; }
23  |   int Semn::GetY () { return Y; }
24  |   void Semn::
          Position (int PosX, int PosY)
25  |   {
26  |     X = PosX; Y = PosY;
27  |   }
28  |   void Semn::Citire ()
29  |   {
30  |    gettext (X, Y, X, Y, &valoare);
31  |   }
32  |   void Semn::Scriere (char Sign)
33  |   {
34  |    Citire ();
35  |    valoare = Sign;
36  |    puttext (X, Y, X, Y, &valoare);
37  |   }
38  |  void Semn::Modificare (char Sign)
39  |   {
40  |    valoare = Sign;
41  |    puttext (X, Y, X, Y, &valoare);
42  |   }
43  |   void Semn::Indicare ()
44  |   {
45  |    puttext (X, Y, X, Y, &valoare);
46  |   }
47  |   void Semn::Stergere ()
48  |   {
49  |    valoare = ' ';
50  |    Indicare ();
51  |   }
52  |  void Semn::
        Miscare (int NewX, int NewY)
53  |   {
54  |    char memo_valoare;
55  |
56  |    memo_valoare = valoare;
57  |    Stergere ();
58  |    valoare = memo_valoare;
59  |    Position (NewX, NewY);
60  |    Indicare ();
61  |   }
62  |  void Semn:
        Copiere :(int NewX, int NewY)
63  |   {
64  |    Position (NewX, NewY);
65  |    Indicare ();
66  |   }
67  |   void Semn::
          Seteaza_attr (char attr_nou)
68  |   {
69  |    attr = attr_nou;
70  |    Indicare ();
71  |   }
72  | };
73  |
74  | // definitia clasei Semn2
75  |
76  | class Semn2 : public Semn
77  | {
78  |  protected:
79  |  char valoare_veche;
80  |  char attr_vechi;
81  |
82  |  public:
83  |  Semn2 (int InitX, int InitY) :
          Semn (InitX, InitY)
84  |  {
85  |   valoare_veche = 0x0;
86  |   attr_vechi = 0x0;
87  |  }
88  |
89  |  // Functii-element publice
90  |
91  |   void Semn2::Citire ()
92  |   {
93  | gettext (GetX(), GetY(), GetX (),
          GetY(),&valoare_veche);
94  |   }
95  |   void Semn2::Scriere (char Sign)
96  |   {
97  |    Semn2::Citire ();
98  |    Semn::Scriere (Sign);
99  |   }
100 |   void Semn2::Stergere()
101 |   {
102 |    valoare = valoare_veche;
103 |    attr = attr_vechi;
104 |    Indicare ();
105 |   }
106 |  void Semn2::
        Miscare (int NewX, int NewY)
107 |   {
108 |    char memo_valoare;
109 |    char memo_attr;
110 |
111 |    memo_valoare = valoare;
          valoare =valoare_veche;
112 | memo_attr =attr;attr= attr_vechi;
113 |    Indicare ();
114 |    valoare = memo_valoare;
115 |    attr = memo_attr;
116 |    Position (NewX, NewY);
117 |    Citire ();
118 |    Indicare ();
119 |   }
120 |  };
121 |
122 |  int main ()
123 |  {
124 |   int X1, Y1, s, z;
125 |
126 |   X1 = 1; Y1 = 1;
127 |   Semn BSemn (X1, Y1);
128 |   BSemn.Scriere ('+');
129 |   BSemn.Seteaza_attr (BLUE);
130 |   for (s = 0; s < 25; s++)
131 |    for (z = 0; z < 80; z++)
132 |     BSemn.Copiere (X1+z, Y1+s);
133 |   X1 = 4; Y1 = 10;
134 |   Semn2 ASemn (X1, Y1);
135 |   ASemn.Scriere ('A');
136 |   ASemn.Seteaza_attr (WHITE);
137 |   printf ("\n Apasati o tasta ... ");
138 |   getch ();
139 |   ASemn.Miscare (X1 + 4, Y1 - 4);
140 |   getch ();
141 |   ASemn.Modificare ('B');
142 |   getch ();
143 |   ASemn.Stergere ();
144 |   getch (); clrscr ();
145 |   return 0;
146 |  }
```

EX5.CPP

poziţie. Această metodă este "Modificare ()". Totodată, cele două obiecte urmează să se deosebească şi prin atributele ecran. Acest lucru se întîmplă prin intermediul metodei "Setează_attr ()".

De fapt, stricto sensu, în acest exemplu sînt două obiecte. Unul din ele acoperă fundalul, fără a necesita informaţie despre ce a fost înainte. Celălat obiect este o clasă derivată, extinsă cu caracteristici de "istorie". Acest lucru determină modificarea necesară a metodelor.

Următorul listing arată cum ar fi dotate două clase diferite cu toate metodele necesare într-o manieră uzuală de programare. De remarcat că toate atributele lui "Semn" au fost declarate publice, astfel încît ele pot fi accesate şi din clasele derivate. Acest exemplu vrea numai să evidenţieze diferenţa faţă de limbajele de programare convenţionale.

Procedeul uzual însemna că la tipuri de date noi trebuia adaptat întotdeauna codul vechi, chiar dacă modificarea tipului datelor nu se răsfrîngea direct asupra anumitor părţi ale programului. Ceea a generat, deseori, surprinzător de multe erori. Scopul ar trebui să fie deci testarea serioasă a codului necesar, urmînd ca acest cod să fie dat mai departe claselor derivate, care moştenesc caracteristicile.

```
1  | // o incercare in modul grafic
2  | // def.: in fisier separat: first.h
3  | // fctiile-element sint in first.cpp
4  |
5  | #include <graphics.h>
6  | #include <stdio.h>
7  | #include <conio.h>
8  |
9  | #include "first.h"
10 |
11 | // Fctiile-element ale cl "Position"
12 |
13 | Position::
     Position (int InitX, int InitY)
14 | {
15 |   X = InitX;
16 |   Y = InitY;
17 | };
18 |
19 | int Position:: GetX(void)
20 | {
21 |   return X;
22 | };
23 |
24 | int Position:: GetY(void)
25 | {
26 |   return Y;
27 | };
28 |
29 | // Fctiile-element ale clasei "Semn"
30 | //  Sistemul grafic a fost initializat
31 | // de catre programul principal
32 |
33 | Semn:: Semn(int InitX,
     int InitY): Position (InitX, InitY) {
34 |   moveto (InitX, InitY);
35 | }
36 |
37 | void Semn::Scrie(char *valoare) {
38 |   moveto(getx(), gety());
39 |   outtext (valoare);
40 | };
41 |
42 | int main()
43 | {
44 |   int midx, midy;
45 |   int gdriver = DETECT, gmode;
46 |   char *text;
47 |
48 |   initgraph(&gdriver, &gmode,
       "\\BORLANDC\\BGI\\");
49 |
50 |   midx = getmaxx() /2;
51 |   midy = getmaxy() /2;
52 |
53 |   Semn APos (midx, midy);
54 |   text = "Hello, world\0";
55 |   APos.Scrie(text);
56 |
57 |   getch();
58 |   closegraph();
59 | }
```

FIRST.CPP

Dacă o metodă nu este satisfăcătoare pentru o clasă, atunci

```
1  | // first.h contine doua clase:
2  | // "Position" da pozitia, ( X si Y)
3  | // "Semn" = caracter in  Position
4  |
5  | // Functiile element sint in first.cpp
6  |
7  | class Position {
8  | protected:
9  | // cl.  deriv. au acces la date "private"
10 |
11 |   int X;
12 |   int Y;
13 |
14 | public:
15 | // Functii ce pot fi apelate din ext.
16 |
17 |   Position (int InitX, int InitY);
18 |   int GetX();
19 |   int GetY();
20 | };
21 |
22 | class Semn : public Position {
23 | // derivata din clasa "Position"
24 | // in int. lui "Semn", X ,Y protejate
25 |
26 | protected:
27 | // Pentru a permite accesul
28 | // functiilor derivate din "Semn"
29 |   char *value;
30 |
31 | public:
32 |   Semn (int InitX, int InitY);
33 |   void Scrie (char *value);
34 | };
```

FIRST.H

numai aceasta trebuie ajustată pe măsura tocmai acestei clase.

Clasa "Semn2" posedă funcţiile-element modificate "Citire ()", "Scriere ()", "Ştergere ()" şi "Mişcare ()". De remarcat în acest exemplu că în metoda Scriere a clasei Semn2 au fost combinate "Citire ()" din clasa Semn2 cu "Scriere" din clasa Semn. În partea a doua a cursului, acest exemplu va fi mai reluat odată, reducînd sursa programului.

Toate exemplele prezentate au în comun faptul că funcţiile lor element sînt funcţii inline. Urmează un exemplu în care în header sînt prezentate numai definiţiile de clase cu metodele aferente, în timp ce descrierea metodelor urmează să se facă în altă parte. Ultimul exemplu, Hello.CPP, afişează un şir de caractere în mod grafic. Important este nu conţinutul exemplului, ci forma: faptul că în fişierul header sînt prezentate numai definiţii.

(I.F.)

Tip obiect: Un tip obiect este un tip de date definit de utilizator. În el sînt definite structurile de date şi metodele de prelucrare corespunzătoare.

Obiect: Un obiect este declarat şi manipulat ca şi o variabilă. Compilatorul rezervă deci pentru un obiect spaţiul de memorie necesar, care este accesibil prin intermediul numelui obiectului.

Metodă: Pentru a putea lucra cu datele declarate în interiorul unui obiect, definiţia lui trebuie să conţină şi declaraţiile funcţiilor şi procedurilor corespunzătoare lor. Tocmai acestea însă sînt metodele.

Sarcină: După stabilirea unui obiect, numele unei metode este utilizat pentru a-i transmite o sarcină acestuia.

Protocol: Totalitatea sarcinilor accesibile unui tip obiect este desemnată ca protocol.

Încapsulare: Cuprinderea împreună a structurilor de date şi a metodelor de prelucrare se numeşte încapsulare. Definirea obiectelor în cadrul unui Unit întăreşte încapsularea.

Moştenire: Tipurile obiect îşi pot transmite toate însuşirile unor noi obiecte. Această modalitate de moştenire elimină necesitatea de a redefini metodele deja definite. Trebuie declarat doar ceea ce s-a schimbat faţă de obiectul "mamă" sau care sînt datele şi metodele care trebuie adăugate.

Ierarhie de obiecte: Deoarece tipurile obiect îşi pot moşteni însuşirile, se poate construi o ierarhie arborescentă a tuturor urmaşilor. Toate tipurile obiect ale acestei ierarhii sînt legate prin "legături de rudenie". De datele şi metodele obiectului rădăcină pot dispune toţi urmaşii.

Novell anunţă:

Netware Lite v1.0

Firma Novell, gigantul reţelelor de calculatoare (în primul rînd personale) lansează pe piaţă versiunea sa de sistem de operare de reţea de tipul "peer to peer".

Aceasta oferă unor utilizatori lipsiţi de pretenţii şi mai puţin dispuşi să facă investiţii substanţiale, posibilitatea de a partaja resursele (aplicaţii, fişiere de date, imprimante, mijloace de comunicaţie etc.) de care dispun. Respectivele resurse se află distribuite pe calculatoare pe care rulează sistemul de operare DOS.

Fiecare din calculatoarele cuplate în reţea poate fi utilizat ca server, client sau ambele. Calculatorul server va partaja resursele sale cu calculatoarele client din reţea; un server poate partaja maxim 25 de resurse cu celelalte calculatoare.

Produsul oferă utilizatorului posibilitatea de a executa rapid operaţiuni specifice reţelei cum ar fi accesul la un disc partajat sau la serviciile unei imprimante. Interfaţa cu utilizatorul se prezintă sub forma unui ecran care cuprinde un sistem de meniuri.

Accesul la resursele partajate ale reţelei este controlat prin intermediul unei scheme de protecţie simple.

Pentru fiecare utilizator se creează un singur cont la nivelul întregii reţele; în felul acesta accesul nu se face la nivelul unei resurse anume ci la totalitatea resurselor din reţea. Gestiunea resurselor este uşurată de faptul că ea se realizează de la oricare din calculatoarele din reţea.

Accesul utilizatorului se efectuează la resursă şi nu la calculatorul care o oferă la un moment dat, apărînd posibilitatea deplasării resursei de pe un calculator pe altul fără ca prin aceasta drepturile de acces ale utilizatorului la resursa respectivă să fie redefinite.

Conceptual, acest aspect, combinat cu cel al accesului la reţea şi nu la calculatoarele din reţea, reprezintă o apropiere de noţiunea de prelucrare distribuită.

Toate programele aferente se găsesc pe o singură dischetă de 1,2M. Procesul de instalare este foarte rapid şi nu necesită cunoştiinţe aprofundate.

Partea de hardware pentru reţea este furnizată de o companie legată de Novell încă din 1984, anume Eagle Technology, o diviziune a lui Anthem Electronics (aceasta din urmă cotată la bursa din New York şi creditată cu un venit de $408 milioane în anul 1990).

Produsul se comercializează sub două forme:

un kit iniţial: NetWare Lite Ethernet Starter Kit, ce cuprinde două plăci interfaţă reţea Ethernet, cablurile şi conectoarele aferente, sistemul de operare NetWare Lite pentru două PC-uri (deci două dischete) precum şi documentaţia aferentă;

un kit pentru un nod cuprinzînd sistemul de operare, placa de reţea cu cablul şi conectoarele aferente, precum şi documentaţia respectivă.

Prin achiziţia starter kit-ului se realizează o reţea de două calculatoare (noduri) după care acesteia i se adaugă cîte un nod, prin achiziţia celuilalt kit.

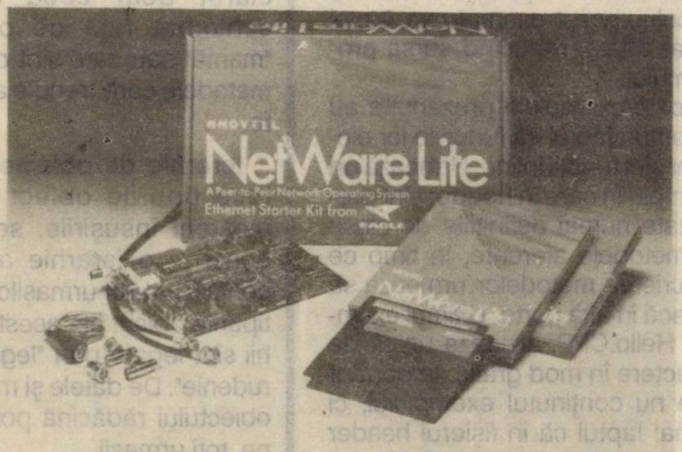
Plăcile de reţea oferite sînt NE2000 (pentru arhitecturi ISA) şi NE/2 (pentru arhitecturi Micro-Channel). Aspectul este extrem de important deoarece dacă peste un timp utilizatorul respectiv doreşte să treacă la o reţea mai sofisticată (NetWare v2.2 sau NetWare v3.11) plăcile de reţea ca şi cablurile vor putea fi folosite fără probleme.

De altfel, NetWare Lite coexistă cu NetWare v2.2 şi v3.11. Produsul (hard + soft) poate fi comandat de la firma SCOP S.R.L., str. Polonă 86, 71154 Bucureşti, tel. 11.74.21, fax 11.73.74.

Firma SCOP S.R.L. este distribuitor Novell pentru România. Sînt binevenite şi solicitări din partea firmelor care doresc să devină remarketeri pentru orice alte produse Novell.

Preţul sistemului de operare este de $99/nod, la care se adaugă costul plăcii de reţea, al cablului şi al conectoarelor aferente.

Marius Sturzoiu , SCOP S.R.L.



EAGLE NetWare Lite Ethernet Starter Kit

NOVELL PRIN scop

NOVELL INTERNATIONAL AUTHORISED REMARKETER

Vă propune integrarea în sistemul autorizat de distribuţie în România.
Veţi beneficia astfel în mod oficial de:

- **programul de upgrade al firmei NOVELL**
- **autorizarea firmei dvs. ca NOVELL AUTHORISED RESELLER**
- **suportul tehnic şi de marketing necesar.**

IBM

EVEREX
EVER for EXcellence

COMPEX

Summagraphics / HOUSTON INSTRUMENT

scop

AUTHORISED REMARKETER IN ROMANIA

Telefon: 11.74.21 / 11.92.48. **Fax: 11.73.74.**

Noi sîntem soluţia !

Expoziţie

În perioada 8-10 octombrie '91, Facultatea de Electrotehnică din Cluj Napoca a fost gazda **expoziţiei de tehnică de calcul** şi a "**Zilelor profesionale SzKI**" organizate de societatea comercială COMPREX import-export S.R.L. din Cluj Napoca şi de SzKI (Institutul de Cercetări şi Centrul de Inovaţii pentru Tehnică de Calcul Budapesta). Au fost prezentate mai multe configuraţii de calculatoare personale (Laptop-uri, AT-uri, 386), la preţuri rezonabile (un AT standard 189.000), soluţii de interconectare, demonstraţii software.

Firma mixtă româno-germano-maghiară COMPREX, distributor autorizat ARL, va comercializa, printre altele, renumitele calculatoare produse de această firmă şi pachetele software Recognita (nr. 3 mondial la categoria software OCR, produs comercializat în 16 state), Ventura Publisher (versiune românească), Pro-C (generator de programe sursă C).

Firma a anunţat şi un **program de amnistie** - în schimbul copiilor ilegale de Ventura Publisher, în următoarele 6 luni, pot fi obţinute copii legale ale acestui produs la cca. 100.000 lei.

Firma COMPREX poate fi contactată la unul din telefoanele: 95-117623, 95-115145, sau prin fax: 95-112474.

Tib '91

Multicolore impresiile de la acest tîrg internaţional, organizat parcă sub aşteptări. Dacă este sau nu exclusiv meritul mineriadei, este discutabil; cert este că o susţinută activitate de instalare/amenajare - încă în a treia zi de tîrg! - este o notă proastă. Tot lipsuri organizatorice sînt cauza faptului că o listă a firmelor ce expun este atît de greu de obţinut încît e preferabil să te laşi păgubaş. De liste tematice - produse chimice, industria lemnului sau calculatoare - să nu mai vorbim.

Indiferente la ce povesteşte televiziunea pe la telejurnal, destule standuri s-au umplut abia pe parcurs - şi nu numai datorită faptului că unele firme au fost anunţate prin telex despre standul "obţinut", trecîndu-se peste "amănuntul" că nu au telex...

Totuşi, faptul că vă putem prezenta alăturat lista celor circa 80 de firme pe care le-am reperat la tîrg denotă o ofensivă continuă a tehnicii de calcul. Am încercat să ne limităm destul de strict la specificul revistei, trecînd (prea repede şi cu o undă de regret) pe lîngă destul de multele firme ce foloseau PC-uri înglobate în aplicaţii industriale sau în aparatură de laborator.

Multe firme comercializează PC-uri, de la XT-uri ieftine-ieftine (ODE-ROM) pînă la AT-uri "supergrele" cu 486, SVGA, harddisk de 180 Mbyte sau chiar mai mult. De la cei care oferă produse de un anumit tip (IBM, Tandon, KT, Laser) şi pînă la angrosişti de genul KSL, ce caută dealeri pentru produsele înşirate într-un catalog-listă de preţuri de 116 pagini tipărite mărunt, mai toate felurile posibile de comerţ sînt reprezentate. Dacă System Plus ţine la "nume" (Dell, Iversson, Summagraphics), SCP se mîndreşte cu cele mai accesibile preţuri. Se profilează cîteva "Computerland"-uri, în care să se poată cumpăra totul dintr-o dată, de la dischete şi mouse-uri pînă la sisteme complete, cu pachete de soft şi documentaţii: MBL Computerland, SCP, ABTech, Circle Systems International, ADISAN, Tokyo Electronics etc. Surprinde plăcut Infocomp, care oferă şi elemente componente: cipuri de memorie, coprocesoare, integrate diverse.

În ce priveşte softul, produsele Borland, vîndute sub amnistie de Logic la preţuri incredibil de avantajoase, stau alături de produsele pentru a căror vînzare este nevoie de autorizaţii speciale, cum ar fi AutoCad 11.0 (ce poate fi cumpărat cu reducere de 70% de la A&C cu condiţia să fie folsit numai pentru învăţămînt şi nu în proiectare!). Reţelele Novell, prezente ca un laitmotiv, sînt "atacate" de Digital, care oferă condiţii avantajoase tuturor posesorilor de I100 sau Coral care, conştienţi de limitele PC-urilor, optează pentru maşini DEC şi VMS sau Unix. Unix-ul vine încet, dar sigur (Teleinvest, Bull), iar concepte revoluţionare precum Open System Architecture pot fi urmărite "pe viu" la MDS GmbH, unde - ca într-un joc cu cuburi - plecînd de la o configuraţie minimă, se pot construi configuraţii extrem de complexe adăugînd module hard şi soft noi, de o manieră cuceritor de simplă.

Periferia se diversifică: pot fi cumpărate imprimante rapide (RCD, FW-20, 2000 linii/minut) sau color (AREXIM), discuri de pînă la 300 Mb (RCD), CD-ROM-uri (Alfa-Bit S.R.L. 600 Mbytes, 150.000 lei), cartele fax sau modem, plottere (HP în primul rînd) etc. Poate cea mai tentantă ofertă: Top9+ vinde benzi magnetice Qualstar, împreună cu softul care permite transferul de date de pe benzi Felix sau mini.

Şi în materie de comunicaţii au apărut noutăţi: Motorola oferă aparatură profesională pentru telecomunicaţii, iar Cedru S.A. anunţă o legătură ISDN cu Germania, via satelit, în colaborare cu ER&P, reprezentanţi unici Eurodata pentru Europa de Est.

Devine grea viaţa cumpărătorului de tehnică de calcul: în orice direcţie se răsteşte la Sesam, uşa se deschide...

Iosif Fettich

Firma	Observaţii	Fax	Telefon
A&C International	Autodesk - Authorised Dealer	535328	535315
ABTech			104273
ADISAN			158165
AREXIM	imprimante		
ARIS Group		595270	595270
ARC Braşov S.R.L.		951/50022	951/50022

ARTIMEX S.R.L.			
ATC - A-Tronic Computer		102428	106196
ATI S.R.L.		959/12425	959/16899
Alfabit S.R.L.	CD-ROM, com. radio	114973	144639
Axaware Publishing	DTP		
Baneasa S.A.	fost IPRS	11203	334050
Baza de instruire informatic	cursuri la Vatra Dornei		988/73534
Blue Ridge International			
BroDeCo	se retrage din calculatoare	111895	112263
Bull	Unix, Zenith		(331)45029090
C.P. Info S.A.		133840	816125
Cedru S.A.	ISDN, Eurodata	433102	435954
Ciel Roumanie	financiar - contabil		
Circle Systems International	soc. romano-americana	143681	143001
DATATIM S.A.		961/33418	961/30078
Delta Design		136040	154580/2404
Digital		69874275	69875700
EDCG		212644	143594
East Europ Trading Ltd.			794194
Electron S.R.L.			
Electronum		120438	120439
Eurotrade			
Eurotrade Company			
FEPER			876714
Finro	în colaborare cu ICI	127556	336999
Goldstar			
HAMEG			
Hewlett Packard		120294	697080
IBM			143929
ICE + HITEC	Unix	450253	450253
ICE Felix Computers S.A.		887820	886030
IIRUC Enter	firmsa romano-elena		
IIRUC S.A. MDS GmbH	OSA, SQL, Posix, C.TOS, Unix		882070/249
IMC Data Systems	soc. romano-olandeza		
IMSAT		376101	376550
INFA S.A.	contabilitate, INFA NET	127501	127501
ITC + ATC			
ITC + AMT	imprimante AMT	797770	797770
Infcon S.R.L.			
Info C.P.	fost C.C. al M.I.Et.		
InfoStar			921/14143
Infocomp	si componente diverse	157795	132697
Innovator			334097
Intercom Leu S.R.L.	tehnica de birou	120317	136209
K.S.L. Europe	en gros; cauta dealer		
KT Technology Romania Ltd.	RCD + Singapore	141022	152316
LECOM	reprez. NEC		506420
Laser + Realsoft			
Logic		757135	754900
M&M Computers			
MBL Computers SRL	Computerland Romania	127568	333227
MDF Europe Ltd.	Pilot Computer Systems	(0483)574217	(0483)65951
Memorii Timisoara			
Mercator			111675
MICROROM	cauta dealer	146114	144309
Motorola			
ODEROM	XT-uri ieftine	142240	151507
RCD			
Robor Chisinau	soc. sovieto-americana	(0422)229321	(0422)222259
RomSoft	fost BNP		652540
Romtrust SRL	imprimante Citizen	334528	336616
Romwest Euro S.R.L.			
SCOP		117374	117421
SCP		753155	757005
SIS			115326
SIX Ltd.	DTP		776595
Shark			396274
SiaSoft S.R.L.	soft spec. ind. alim.	150439	148797
Sistema S.R.L.		795970	795970
Software ITC SA		127612	127611
Sogetec	inf. industriala		
Sysgraph	DTP, Racall Redac		
System PLUS S.R.L.	reprez. IBM, Dell, Iversson,..		791391
Tandon Romania S.R.L.		171039	170960
Tecom Multicom	timp real - sere, dozatoare,..	752052	387814
Teleinvest	Unix, Progress, 3COM	565678	565678
Tokyo Electronics		654510	657030
Top 9+	labcards; benzi magnetice	107036	102787
Trend S.R.L.	dealer HP	333389	127634

```
#include <string.h>

#define L_PASS 10

void change_password(char *new_pass);

/* password0 - parola initiala
   password  - parola de lucru
   newpass   - noua parola
   exe_name  - numele executabilului */
char password0[L_PASS+1]="CUCUCUCUCU",
     newpass[L_PASS+1]="testtestte",
     password[L_PASS+1],
     exe_name[50];
/*-------------------------------------------------------------------*/
void main(int argc, char *argv[])
{
     strcpy(password, password0);
     strcpy(exe_name, argv[0]);
     strcpy(newpass, argv[1]);
     printf("Parola curenta este: %s\n", password);
     printf("Parola noua va fi: %s\n", newpass);
     change_password(newpass);
}
/*-------------------------------------------------------------------*/
void change_password(char *new_pass)
{ static unsigned
     file_handle, psp_seg, pass_seg, pass_off, l_header;
     static unsigned long exe_pos;

     /* det. dist. rel. a parolei fata de inceputul programului
        incarcat in memorie. Se tine seama de lung. PSP */
     asm {
          mov ah, 62h
          int 21h
          mov word ptr psp_seg, bx          //segment PSP
          mov pass_seg, SEG password0
          mov pass_off, OFFSET password0
     }
     exe_pos=16*(pass_seg-psp_seg)+pass_off-0x100;

     /* determina marime header fisier pe disc */
     asm {
          mov ax, SEG exe_name  //deschidere fisier
          mov ds, ax
          mov dx, OFFSET exe_name
          mov al, 2
          mov ah, 3dh
          int 21h
          mov file_handle, ax
          mov al, 0          //pozitionare in fisier
          mov bx, file_handle
          mov cx, 0          //cx:dx offs. in fis. a dim. header
          mov dx, 8h
          mov ah, 42h
          int 21h
          mov ax, SEG password  //cit. marime header fisier
          mov ds, ax
          mov dx, OFFSET password
          mov bx, file_handle
          mov cx, 2
          mov ah, 3fh
          int 21h
          mov ax, word ptr password    //manevra
//        mov ah, byte ptr password+1
          mov l_header, ax
     }

     /* det. dist. relativa a parolei fata de incep. fis.
        executabil pe disc */
     exe_pos+=16*l_header;

     /* modifica parola */
     strcpy(password, new_pass);
     asm {
          mov al, 0          //pozitionare in fisier
          mov bx, file_handle
          mov cx, word ptr exe_pos+2   //cx:dx poz. in fisier
          mov dx, word ptr exe_pos
          mov ah, 42h
          int 21h
          mov ax, SEG password  //scriere parola
          mov ds, ax
          mov dx, OFFSET password
          mov bx, file_handle
          mov cx, L_PASS
          mov ah, 40h
          int 21h
          mov bx, file_handle   //inchidere fisier
          mov ah, 3eh
          int 21h
     }
}
/*-------------------------------------------------------------------*/
```

PAROLVAR.C

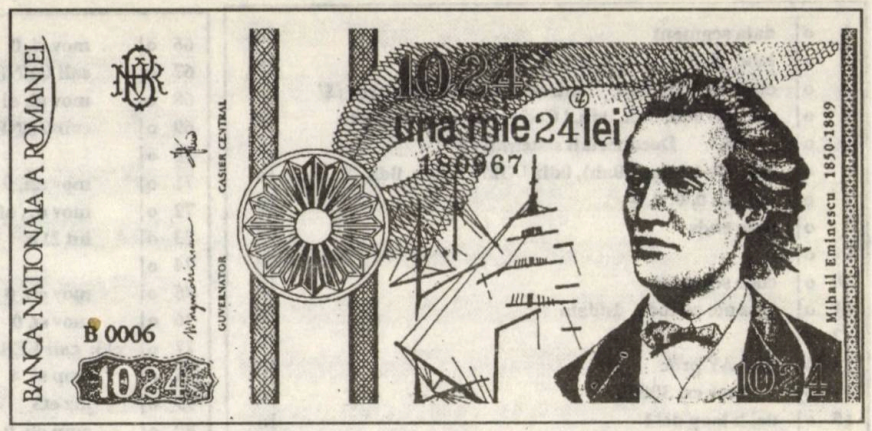
Parolă variabilă

Implementarea unei parole în cadrul unui program este un lucru destul de simplu. Este suficientă crearea unei constante şir cu o valoare cunoscută numai de anumiţi utilizatori şi limitarea accesului la execuţia programului pe baza comparării unui string introdus de utilizator cu stringul predefinit. Lucrurile se complică atunci cînd se doreşte modificarea parolei într-un mod interactiv, adică evitîndu-se modificarea fişierului sursă şi recompilarea programului. Pentru rezolvarea acestei probleme propunem folosirea funcţiei change_password () prezentată în acest mic articol. Funcţia a fost testată în Turbo C, dar autorii vă asigură de portabilitatea sa pentru orice implementare de limbaj ANSI C pe calculatoare compatibile IBM-PC sub sistemul de operare MS-DOS. Prezentăm şi cadrul general în care funcţia a fost testată.

George Anescu, Pandele Stănescu,
I.C.N. Piteşti

1024

Conform promisiunii din numărul 7 al revistei, vom premia - în fiecare număr al revistei - cel mai interesant program **mai mic de 1024 octeţi** (în sursă) pe care îl primim spre publicare. Primul cîştigător: **dl. Buleu Daniel,** str. Simeon Florea Marian nr. 3, 5875 Rădăuţi, pentru programul DINAMIC.ASM. Felicitări !



Cite ceva despre tastatura AT

Tastatura AT (şi microcontrollerul incorporat) are facilităţi de programare şi este prin aceasta mult mai interesantă decît vechile tastaturi model PC. Informaţii relativ detaliate relativ la aceste facilităţi pot fi găsite de ex. în (binecunoscutul) TechHelp (The Electronic Technical Reference Manual, Copyright (c) 1985, 1990 by Flambeaux Software, Inc. Cele două programe de mai jos sînt două exemple de utilizare. Ele trebuisc introduse cu un editor ASCII, asamblate cu un macroasamblor şi link-editate.

DINAMIC.EXE se va juca cu LED-urile corespunzătoare tastelor "Lock", transformîndu-le în lumini dinamice.

PAROLA.EXE poate fi utilizat pentru protecţia accesului în sistemul Dvs. de calcul. Introducînd o linie cu numele fişierului în AUTOEXEC.BAT veţi fi puşi în situaţia de a tasta o parolă pe care nici Dvs. nu o ştiţi! Vă vin în ajutor LED-urile amintite care, în caz că sînt în ordinea NUM, CAPS, SCROLL (de la stînga la dreapta) vă vor da codul binar al următoarei taste de apăsat ("on"=1, "off"=0). Dacă ordinea nu este aceasta, trebuie doar să modificaţi tabela de conversie "tabc".

La ieşirea din program se reface starea LED-urilor în concordanţă cu octetulde stare al tastaturii de la adresa 0000:0417.

Buleu Daniel,
str. Simeon Florea Marian nr.3,
5875 Rădăuţi, jud. Suceava

```
1  o|  data segment                                          |o
2  c|  etc  db 0ah, 0dh, '   Press any key to stop ! $'     |o
3  o|  data ends                                            |o
4  o|                                                       |o
5  o|  code segment                                         |o
6  o|  assume cs:code, ds:data                              |o
7  o|                                                       |o
8  o|  DELAL proc                                           |o
9  o|      mov cx, 2000h                                    |o
10 o|  del2: loop del2                                      |o
11 o|      ret                                              |o
12 o|  DELAL endp                                           |o
13 o|                                                       |o
14 o|  DELAY proc                                           |o
15 o|      mov dl, 40h                                      |o
16 o|  del:  call DELAL                                     |o
17 o|      dec dl                                           |o
18 o|      jnz del                                          |o
19 o|      ret                                              |o
20 o|  DELAY endp                                           |o
21 o|                                                       |o
22 o|  PROG  proc                                           |o
23 o|      mov al, 0edh                                     |o
24 o|      out 60h, al                                      |o
25 o|      call DELAL                                       |o
26 o|      ret                                              |o
27 o|  PROG  endp                                           |o
28 o|                                                       |o
29 o|  start:     mov ax, data                              |o
30 o|      mov ds, ax                                       |o
31 o|      mov ax, 0                                        |o
32 o|      mov es, ax                                       |o
33 o|      mov ah, 9                                        |o
34 o|      mov dx, offset etc                               |o
35 o|      int 21h                                          |o
36 o|                                                       |o
37 o|  begin:     call PROG                                 |o
38 o|      mov al, 2                                        |o
39 o|      out 60h, al                                      |o
40 o|      call DELAY                                       |o
41 o|      call PROG                                        |o
42 o|      mov al, 1                                        |o
43 o|      out 60h, al                                      |o
44 o|      call DELAY                                       |o
45 o|      call PROG                                        |o
46 o|      mov al, 4                                        |o
47 o|      out 60h, al                                      |o
48 o|      call DELAY                                       |o
49 o|      mov ah, 0bh                                      |o
50 o|      int 21h                                          |o
51 o|      cmp al, 0                                        |o
52 o|      jz begin                                         |o
53 o|                                                       |o
54 o|  quit: mov al, 0edh                                   |o
55 o|      out 60h, al                                      |o
56 o|      mov si, 417h                                     |o
57 o|      mov al, es:[si]                                  |o
58 o|      mov cl, 4                                        |o
59 o|      shr al, cl                                       |o
60 o|      and al, 7                                        |o
61 o|      call DELAL                                       |o
62 o|      out 60h, al                                      |o
63 o|      mov ah, 04ch                                     |o
64 o|      int 21h                                          |o
65 o|                                                       |o
66 o|  code ends                                            |o
67 o|  end start                                            |o
```

DINAMIC.ASM

```
1  o|  data segment                                          |o
2  o|  tabc  db 0, 1, 4, 5, 2, 3, 6, 7                       |o
3  o|  ecod  db 0ah, 0ah, '  Introduceti codul de acces : $' |o
4  o|  egr   db 0ah, 0ah, 0dh, ' Cod incorect !', 0ah, 0dh   |o
5  o|       db '     Deconectati sistemul ! $'               |o
6  o|  eok   db 10 dup (0ah), 0dh, ' Ok!', 7, 0ah, 0dh, '$' |o
7  o|  loc   db 0, 0, 0, 0                                   |o
8  o|  data ends                                            |o
9  o|                                                       |o
10 o|  code segment                                         |o
11 o|  assume cs:code, ds:data                              |o
12 o|                                                       |o
13 o|  DELAY proc                                           |o
14 o|      mov cx, 3000h                                    |o
15 o|  del2: loop del2                                      |o
16 o|      ret                                              |o
17 o|  DELAY endp                                           |o
18 o|                                                       |o
19 o|  GENER proc                                           |o
20 o|      mov ah, 0                                        |o
21 o|      mov bl, 16                                       |o
22 o|      div bl                                           |o
23 o|      and ah, 7                                        |o
24 o|      mov loc[si], ah                                  |o
25 o|      inc si                                           |o
26 o|      and al, 7                                        |o
27 o|      mov loc[si], al                                  |o
28 o|      inc si                                           |o
29 o|      ret                                              |o
30 o|  GENER endp                                           |o
31 o|                                                       |o
32 o|  VERIF proc                                           |o
33 o|      mov al, 0edh                                     |o
34 o|      out 60h, al                                      |o
35 o|      call DELAY                                       |o
36 o|      mov al, loc[si]                                  |o
37 o|      out 60h, al                                      |o
38 o|      mov ah, 0                                        |o
39 o|      mov di, ax                                       |o
40 o|  et1: mov ah, 7                                       |o
41 o|      int 21h                                          |o
42 o|      cmp al, 30h                                      |o
43 o|      jb et1                                           |o
44 o|      cmp al, 39h                                      |o
45 o|      ja et1                                           |o
46 o|      mov dl, tabc[di]                                 |o
47 o|      sub al, 30h                                      |o
48 o|      cmp dl, al                                       |o
49 o|      je et2                                           |o
50 o|      mov ch, 1                                        |o
51 o|  et2: inc si                                          |o
52 o|      mov ah, 2                                        |o
53 o|      mov dl, '.'                                      |o
54 o|      int 21h                                          |o
55 o|      ret                                              |o
56 o|  VERIF endp                                           |o
57 o|                                                       |o
58 o|  start:  mov ax, data                                 |o
59 o|      mov ds, ax                                       |o
60 o|      mov ax, 0                                        |o
61 o|      mov es, ax                                       |o
62 o|                                                       |o
63 o|      mov ah, 2ch                                      |o
64 o|      int 21h                                          |o
65 o|      mov al, dh                                       |o
66 o|      mov si, 0                                        |o
67 o|      call GENER                                       |o
68 o|      mov al, dl                                       |o
69 o|      call GENER                                       |o
70 o|                                                       |o
71 o|      mov ah, 9                                        |o
72 o|      mov dx, offset ecod                              |o
73 o|      int 21h                                          |o
74 o|                                                       |o
75 o|      mov ch, 0                                        |o
76 o|      mov si, 0                                        |o
77 o|  et3: call VERIF                                      |o
78 o|      cmp si, 4                                        |o
79 o|      jnz et3                                          |o
80 o|      cmp ch, 0                                        |o
81 o|      jz et5                                           |o
82 o|      mov ah, 9                                        |o
83 o|      mov dx, offset egr                               |o
84 o|      int 21h                                          |o
85 o|  et4: mov ah, 2                                       |o
86 o|      mov dl, 7                                        |o
87 o|      int 21h                                          |o
88 o|      call DELAY                                       |o
89 o|      jmp et4                                          |o
90 o|  et5: mov ah, 9                                       |o
91 o|      mov dx, offset eok                               |o
92 o|      int 21h                                          |o
93 o|                                                       |o
94 o|  quit: mov al, 0edh                                   |o
95 o|                    out 60h, al                        |o
96 o|      mov si, 417h                                     |o
97 o|      mov al, es:[si]                                  |o
98 o|      mov cl, 4                                        |o
99 o|      shr al, cl                                       |o
100 o|     and al, 7                                        |o
101 o|     call DELAY                                       |o
102 o|                    out 60h, al                       |o
103 o|     mov ah, 04ch                                     |o
104 o|     int 21h                                          |o
105 o|                                                      |o
106 o|  code ends                                           |o
107 o|  end start                                           |o
```

O glumă

Dacă sînteţi în situaţia de a folosi calculatorul cu care lucraţi împreună cu alţi colegi, este aproape sigur că fiecare are cerinţele lui proprii vizavi de acesta. Poate lucrează cu alt driver de tastatură, poate preferă alt mesaj de prezentare la lansare, poate are nevoie de nişte TSR-uri care pe Dvs. vă deranjează, ocupînd prea mult loc, etc. Şi care este locul ideal de a-şi "croi" un sistem după gust, dacă nu AUTOEXEC.BAT? Iar dacă gusturile diferă substanţial, scandalul tinde să se permanentizeze. De aici ideea de a crea un AUTOEXEC.BAT "nemodificabil". Trucul este surprinzător de simplu, dacă dispuneţi de un program care să permită editarea unui fişier COM (PC Tools, NU, etc.).

Mai întîi, confecţionaţi-vă o dischetă sistem care să nu conţină numai fişierele sistem, ci şi alte programe utilitare, cum ar fi FORMAT, RECOVER, CHKDSK, KEYB etc. După care creaţi un nou fişier AUTOEXEC, căruia însă e bine bentru început să-i daţi alt nume - GLUMA.BAT de ex.

Selectaţi apoi de pe discheta unul din programele utilitare pe care în mod normal nu îl utilizaţi - RECOVER.COM de ex. Vă notaţi ora şi data de pe acesta, aşa cum le afişează DIR-ul. Acum ştergeţi RECOVER.COM, după care redenumiţi GLUMA.BAT în RECOVER.COM. Daţi acum noului fişier ora şi data celui original (cu TOUCH, de ex.). La prima vedere, nimeni nu va putea descoperi că acest fişier este de fapt un fişier de comenzi. Aceste pregătiri sînt premisele pentru trucul propriu

-zis. Intraţi acum cu un editor de discheta în COMMND.COM. În MS-DOS v3.3 găsiţi în sectorul relativ 11 înregistrarea "AUTOEXEC.BAT". Schimbaţi acum pur şi simplu această valoare în RECOVER.COM. Puteţi mări confuzia: ştergeţi AUTOEXEC-ul original, după care daţi comanda

copy recover.com autoexec.bat

Aveţi acum un nou AUTOEXEC.BAT şi, dacă mai corectaţi şi data şi ora, totul arată aproape ca la început. Diferenţa faţă de discheta originală este aproape imperceptibilă.

Colegului Dvs. care îşi face de lucru la PC, mesajul de eroare îi va produce o sperietură serioasă. Dacă i se pare că va putea rezolva problema modificînd, ca de atîtea ori, AUTOEXEC.BAT-ul, va avea o surpriză. Căci orice modificări va face - mesajul de eroare se va încăpăţîna să apară. Mai ales "specialiştii" de circumstanţă vor fi îmbrobodiţi.

Încă o observaţie: procedeul funcţionează numai în MS-DOS. DR-DOS 5.0 nu se lasă dus: sistemul se blochează pur şi simplu.

(I.F.)

```
1   @@echo off
2   rem GLUMA.BAT
3   cls
4   echo Supervizorul sistemului anunta:
5   echo.
6   echo CATASTROFA ! Harddisk distrus !
                Pericol de incendiu !
7   echo.
8   echo Harddiskul se formateaza ACUM
9   echo.
10  echo Apasati o tasta oarecare si cautati-
11  echo URGENT pe dl. Director Popescu !
12  echo E singurul autorizat sa actioneze!
13  echo in aceste circumstante ! URGENT !
14  echo.
15  pause nul
```

GLUMA.BAT

Recondiţionarea riboanelor pentru imprimantă

Imprimantele PC-urilor au banda tuşată montată într-o cutie (cartuş) mobilă sau fixă. După uzare, dacă avem la dispoziţie un cartuş nou, îl schimbăm. Dacă nu, îi schimbăm numai banda tuşată. Majoritatea cartuşelor au banda tuşată continuă, cu capetele răsucite la 180 de grade înainte de lipire (banda Mobius). Aşadar există trei categorii de benzi tuşate, respectiv cartuşe în care se montează:

□ - nerăsucite
□ - răsucite pe dreapta
□ - răsucite pe stînga

Prima categorie utilizează, în general, neeficient banda întrucît se imprimă pe o singură pistă (centrală) în timp ce celelalte două folosesc două piste. În multe cazuri banda tuşată scoasă dintr-un cartuş care nu foloseşte banda răsucită poate fi folosită în alt cartuş (pentru altă imprimantă) în care este nevoie de răsucire, şi viceversa, bineînţeles cu relipirea corespunzătoare a capetelor.

Chiar banda tuşată uzată mai poate fi încă folosită, retuşînd-o. Acest lucru presupune pregătirea unui amestec de 2/3 tuş ştampilar şi 1/3 glicerină tehnică. Amestecul se pune într-un vas de plastic (butelie de şampon de exemplu) căruia i se ataşează la dop un tub umplut cu împîslitură (din interiorul unui creion carioca). Tuşarea presupune parcurgerea întregii benzi şi "vopsirea" ei cu "pensula" astfel fabricată. O metodă mai rapidă constă în impregnarea unei bucăţi de pîslă cu amestec pregătit (tuş + glicerină) şi tragerea întregii benzi prin această pîslă (eventual printr-o tăietură practicată). În ambele cazuri nu este nevoie ca banda tuşată să fie scoasă afară din cartuş. Pentru a evita neplăcerile legate de contactul cu soluţia, murdărirea mîinilor, se poate folosi o pereche de mănuşi de bucătărie sau chirurgicale. După terminarea operaţiei se introduce banda din nou în cartuş învîrtind de axul antrenor.

Dacă dispunem de bandă nouă, nelipită, o vom introduce în cartuş, fără să-l desfacem, astfel:

- ● - se trage puţin banda veche afară (cca 20 cm) şi se taie la mijloc.
- ● - se lipeşte capătul dinspre axul de antrenare al benzii vechi de un capăt al benzii noi (cum se lipeşte se va vedea mai jos).
- ◐ - se trage capătul nelipit al benzii uzate pînă se termină.
- ● - se învîrteşte de axul de antrenare pînă cînd rămîne afară o bucată de cca 20 cm din banda nouă.
- ● - se trage de banda uzată pînă cînd apare lipitura continuînd tragerea apoi pînă cînd se scoate afară cca 20 cm din banda nouă.
- ● - se rupe la lipitură şi se îndepărtează banda veche.
- ● - se lipesc capetele benzii noi.

Trebuie avut în vedere ca banda nouă să nu fie mai lată decît cea veche întrucît se produc înţepeniri. Înţepeniri se pot produce şi dacă banda nouă se înghesuie (nu încape) în cartuş. Acest lucru se poate datora fie faptului că este mai lungă fie faptului că este mai groasă. În ambele cazuri se mai scurtează.

Chiar dacă dispunem de banda lipită şi răsucită corespunzător recomandăm dezlipirea ei şi utilizarea metodei de mai sus care are avantajul că prelungeşte durata de viaţă a cartuşului (desfacerea repetată duce la deteriorări). Este însă nevoie de o anume perfecţionare în lipire.

Introducerea în cartuş a unei benzi gata lipite se poate face şi prin desfacerea cartuşului. Cartuşul se desface şi dacă se constată că în interiorul lui s-au produs nereguli. Montarea noii benzi într-un cartuş desfăcut se face fără a ne pune problema "plierii" acesteia. Banda se trece printre cele două roţi dinţate antrenoare (dacă roata presoare are arc se împinge puţin în ea), se merge cu ea prin locurile pe unde trebuie să treacă (eventual se răsuceşte cu 180 de grade în sens corespunzător dacă configuraţia cartuşului o cere) şi se scoate afară. Se pune capacul cartuşului la loc şi apoi se trage banda înăuntru prin învîrtirea axului antrenor cu mîna.

În final vă vom prezenta un procedeu eficient de lipire a capetelor tuşate, cu minimum de instrumente :

» 1) un pistol de lipit electric
» 2) o lamă uzată
» 3) o bucată de geam (10x10cm este suficient) aşezată pe o masă. Poate lipsi dacă masa este acoperită cu geam sau melamină.
» 4) o riglă metalică sau din lemn (nu neapărat gradată).

Procesul se desfăşoară astfel :

° - se suprapun feţele interioare (cele dinspre cartuş) ale capetelor benzii astfel încît cele două capete să le ţinem cu mîna dreaptă iar cartuşul cu mîna stîngă:
° - se aşează capetele suprapuse pe geamul de pe masă.
° - se roteşte capătul de deasupra al benzii tuşate, în plan orizontal, în sens trigonometric cu cca 120 de grade (nu este esenţial unghiul, lipitura va ieşi mai mult sau mai puţin înclinată).
° - se aşează rigla peste bandă astfel încît marginea ei să treacă prin punctele de intersecţie exterioare ale benzii (A, B):
° - între punctele A şi B se trece vîrful încălzit al pistolului de lipit pînă cînd se taie complet banda, capetele din dreapta îndepărtîndu-se.
° - va rezulta o lipitură care mai are un anumit "grad" de bavură. Se aşează banda desfăşută, cu bavura în sus iar peste bavură se aşează lama. Cu pistolul de lipit "se calcă" banda peste lamă (ca şi cu fierul de călcat), în lungul bavurii. Va rezulta o lipitură netedă.

(ing. Sava Stan , Buzău)

IMPORTANT !

Începînd cu numărul viitor al revistei, oferim tuturor celor interesaţi posibilitatea de a face

reclamă gratuită

produselor proprii - soft sau hard. Vom disponibiliza în acest scop una sau două pagini, în care vom publica gratuit textele publicitare care ne vor fi trimise. Singura condiţie este ca produsele să fie **elaborate** de cei care-şi fac publicitate. Ne gîndim în primul rînd la diversele aplicaţii, programe, biblioteci etc. care ar merita să fie cunoscute, dar cărora în general autorii lor nu-şi pot permite să le facă publicitatea pe care, poate ar merita-o. Nu putem promite că vom respecta toate posibilele dorinţe relativ la modul de apariţie al acestor reclame, (reclama plătită este cea la care clientul decide cum anume apare) dar putem promite că vom încerca să le publicăm pe toate (chiar dacă va fi 'înghesuială').

REDUCERE !!

Oferim tuturor celor interesaţi posibilitatea de a se abona la revista noastră la un preţ mai convenabil: oricine solicită un număr de cel puţin 5 exemplare, va avea o reducere de 10%; expedierea se va face prin ramburs, iar plata expediţiei o vom suporta noi. Procentul de reducere poate creşte pînă la 20%, la cantităţi ce depăşesc 200 de exempl..

Pentru cei care doresc eventual numere mai vechi ale revistei: mai sînt disponibile o parte din revistele editate anul acesta. Pentru numerele 1-5/91, oferim o reducere de 20%, pentru cei care pot ridica revistele de la redacţie; pentru cei care le solicită prin ramburs, din bonificaţie vom scade taxa de expediţie.

Poşta redacţiei

"...Aş cumpăra cu plăcere revista dumneavoastră, dacă s-ar găsi şi prin chioşcurile din Iaşi, şi dacă ar costa, să zicem, 12 - 15 lei. (Pentru a reduce preţul revistei vă sugerez să folosiţi caractere mai mărunte. Pasionaţii de informatică sînt în majoritate tineri şi au vederea bună. Oricum, o lupă costă mai puţin decît două numere de if)."

Cristian Trelea, Iaşi

Aveţi, desigur, dreptate: revista este scumpă. Ştim acest lucru, dar, cu toate eforturile noastre, mai ieftin nu reuşim să o scoatem. Ne pare foarte rău că am fost nevoiţi să mai ajustăm odată preţul revistei, şi din păcate în sus. Anunţăm însă pe această cale că **pentru toţi abonaţii, preţul revistei rămîne cel vechi, pe durata pentru care a fost achitat abonamentul** (pentru abonaţii care au făcut abonament pînă în data de 30.09.91. Cei mai noi sînt rugaţi să corecteze sumele achitate, dacă sînt în situaţia de a fi făcut abonamentul în perioada în care a intervenit noua scumpire). Este, aceasta, nu atît o modalitate de a încerca să cîştigăm abonaţi noi, cît mai mult o tentativă de a le mulţumi celor vechi. Mai încercăm, desigur, diverse variante, şi sîntem recunoscători oricui ne dă idei viabile în acest sens. Relativ la un caracter de literă mai mic: aşteptăm eventuale păreri în acest sens şi de la alţi cititori. Oricum, nu credem că cititul cu lupa poate face plăcere. Şi am vrea ca revista să fie citită şi cu plăcere.

Mica publicitate

Firma "LEXINA" - S.R.L. vă asigură **service calificat** pentru orice unitate de discuri - **MD50, MD40** şi în special pentru unităţile **MX200.** Tel. 90/43.32.97 - Bucureşti.

Cumpăr documentaţie pentru Turbo Assembler. Bîscoveanu Costel, str. Baba Novac nr. 117, 8700 Constanţa.

Cumpăr en gros dischete 3,5" sau 5,25" HD, firme cunoscute: TDK, Sony, 3M etc. Reţin doar oferte cu preţ. Bucur Daniel, 3125 Mediaş, O.P.1, C.P. 47, jud. Sibiu.

Vînd două microcalculatoare Junior 80B cu unităţi floppy de 5,25" absolut noi. Tel. 952/15768, int. 262

"...Ziceam că revista place. Nu chiar într-atît încît s-o cumpere, totuşi... De vină este şi preţul ridicat, dar şi lipsa de publicitate. Evident, şi publicitatea costă, dar fără, nu faci nimic.

Iar tirajul este insuficient. Cu un tiraj mai mare s-ar putea reduce preţul, care, aşa cum am spus, este unul din obstacolele majore în calea sporirii vînzărilor. Ca şi apariţia neregulată. Ştiu, nu e uşor, dar cercul vicios trebuie rupt.

Ar fi bine să fie şi articole proprii, adică despre soft românesc, sau realizări (periferice, sisteme, reţele) de la noi. Nu-s suficiente doar traducerile sau prelucrările. ...

Ar mai trebui şi nişte comparaţii cu tabele sinoptice... Comparaţiile ar trebui să trateze atît elemente de hard şi sisteme (precum "Oglinda preţurilor" din Germania din mai 1991, "Noutăţi pe piaţa românească" din nr. 4, etc., cît şi soft şi programe de aplicaţii. Astfel ca cine vrea să-şi extindă sistemul să-şi cumpere un PC sau să-şi instaleze o reţea să fie foarte interesat să apeleze dacă nu la Micro ATCI măcar la colecţia revistei "if" pentru ca, pornind de la condiţiile specifice şi necesităţi, sau, mai adesea la noi, de la limitele de buget, să ştie ce să-şi ia.

Un exemplu concret. Un colectiv din institut are un Felix XT, cu harddisk de 20 Mo şi 2 floppy de 360 ko, afişaj grafic CGA. Pentru a mai scoate un ban, vor să se apuce de DTP. Ce program ar fi mai eficient, Ventura, Page Maker sau partea de DTP de la Works? ...

M-ar interesa ceva despre WAN (şi-n if, şi-n documentaţie am mult mai puţin ca LAN) şi despre sistemul de operare al lui Apple (Mac). Şi-au luat unii şi nu ştiu să-l folosească.

Cristian Malide, Bucureşti

Mulţumim, atît pentru aprecieri, cît mai ales pentru sugestii şi observaţii. Ne vom strădui să ţinem seama de ele. Cît despre tiraj şi preţ ... ca şi cum i-ai spune celui ce se-neacă că ar fi mai bine să înoate...

Iosif Fettich

Şi Dumneavoastră puteţi contribui la îmbunătăţirea conţinutului revistei ! Ajutaţi-ne, completînd chestionarul alăturat, şi revista se va apropia mai mult de ceea ce aşteptaţi de la o revistă destinată utilizatorilor de PC-uri ! Vă mulţumim !

Chestionar

1. Cel mai mult mi-au plăcut următoarele articole:
   1. _____
   2. _____
   3. _____
2. Cel mai puţin mi-au plăcut următoarele articole:
   1. _____
   2. _____
   3. _____
3. În numerele următoare aş dori să fie incluse articole care să trateze:
   1. _____
   2. _____
   3. _____
4. Utilizez un calculator:
   |_| acasă |_| la birou
5. Tipul calculatorului utilizat:
   _____
6. Lucrez mai frecvent cu următoarele pachete software:
   1. _____
   2. _____
   3. _____
7. P.C-urile mă interesează ca:
   |_| profesionist |_| hobby
   |_| student |_| elev
   |_| utilizator |_| posibil utilizator
8. Citesc revista "if":
   |_| regulat |_| ocazional
9. Ce-ar mai fi de spus ? _____

Data: _____ Semnătura: _____ if 8 / 91

Informaţii la zi din lumea calculatoarelor personale puteţi obţine numai citind regulat revista "if" !

Micro ATCI
C.P. 64, O.P. 1
RO - 4300 Tîrgu Mureş

Aveţi ceva de vîndut? Doriţi să cumpăraţi ceva ? Vreţi să vă oferiţi serviciile sau aveţi nevoie de ajutor într-o problemă ? Folosiţi mica publicitate de specialitate din "if" !

Micro ATCI
C.P. 172, O.P. 1
RO - 4300 Tîrgu Mureş

Talon de comandă

Doresc să-mi trimiteţi |_____| exemplare ale revistei **if**, pe o perioadă de |_____| apariţii, pe adresa:
Nume&prenume .....
Strada: ..... Nr.: ..... Bloc: ..... Ap.: ..... Tel.: .....
Cod: ..... Localitatea: ..... Judeţ (sector) .....

Preţul unui exemplar este 69 lei. Preţul unui abonament se va calcula cu formula: 69 lei * nr. exemplare * nr. apariţii. Banii se vor expedia prin mandat poştal pe adresa: Micro ATCI, C.P. 64, O.P. 1, 4300 Tîrgu Mureş, Cont nr.: 40729-96010402 Banca Română pt. Dezvoltare Mureş.

if 8 / 91 **Data: ..... Semnătura .....**

Talon pentru mica publicitate

Vă rog să inseraţi în numărul viitor al revistei **if** următorul anunţ:

|_|_|_

OFERTA DE TOAMNA LA AbMod

Dupa marele succes inregistrat in aceasta vara
ACTIUNEA EGA continua :
SISTEME COMPATIBILE IBM PC/AT la numai

239.000 lei

AT-286 : 12 MHZ, 1MB RAM, HD 40 MB, 1 FDD 1,2 MB 5.25", 1 FDD 1.44 MB 3.5",
Monitor color EGA, 1 Mouse, 10 dischete 5.25", 10 dischete 3.5",
Tastatura 101 taste (USA), Interfata seriala, Interfata paralela .

GARANTIE 1 AN *de la livrare !*

Pentru noul dvs. calculator firma AbMod va ofera posibilitatea achizitionarii celor mai noi versiuni de produse software (originale, legal inregistrabile) :

MS-DOS v. 5.0	**27.000 lei**
WINDOWS v. 3.0	**24.600 lei**
MS-DOS v. 5.0 + WINDOWS v. 3.0	**47.600 lei**
NORTON UTILITIES v. 6.0	**63.000 lei**
NORTON COMMANDER v. 3.0	**43.200 lei**
STACKER v. 1.10	**55.800 lei**

(dubleaza capacitatea discului Winchester)

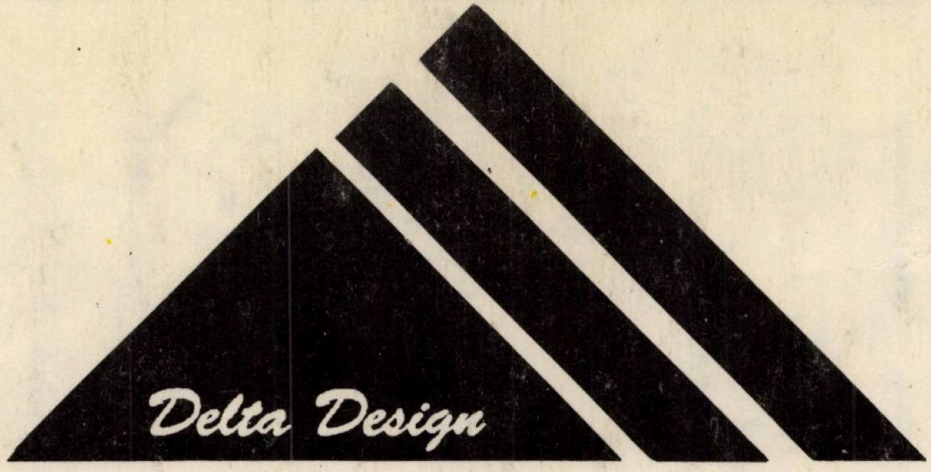
ATENTIE !
Pentru comenzi en-gros firma acorda reduceri de pret .

ASTEPTAM COMENZI FERME LA UNA DIN ADRESELE :

3400 CLUJ-NAPOCA B-DUL 22 DECEMBRIE NR135 TELEFON 95/156350

3700 ORADEA STR DACIA NR. 40 TELEFON 991/60278; FAX 991/56881

Tiparul executat la S. C. TIPOMUR C-da nr. 88/1991 Tiraj: 9000 ex.



Delta Design S.A., Hotel Bucureşti, 63-81 Victoria Street,
Wing D - Agencies, Stair E 2, 4th Floor, Sect. 1, Bucharest,
Phones: 15.42.26 / 40 015.45.80 Ext.2404 / Fax: 40 13.60.40.

99.94% of the users fully satisfied of the delivered products

Authorised distributors for the following companies:

Tulip Computers, Apple Computer Inc., Microsoft, Autodesk,
Ashton Tate, Claris, Corel, Delta Design International etc.